

# Anexo I: Plan de proyecto software

## *Simulador del protocolo de control de enlace de datos de alto nivel (HDLC)*

**Trabajo de Fin de Grado**

**INGENIERÍA INFORMÁTICA**



**Julio de 2023**

**Autor**

*Raúl Melgosa Salvador*

**Tutores**

*Ángeles M<sup>a</sup> Moreno Montero*

*Sergio Bravo Martín*

# Índice de contenido

Introducción .....	4
Estimación del esfuerzo .....	5
Complejidad de los actores (UAW) .....	6
Complejidad de los casos de uso (UUCW) .....	7
Factores de complejidad técnica (TCF) .....	10
Factores de complejidad del entorno (ECF).....	13
Cálculo de la estimación del esfuerzo ( <i>EZEstimate</i> ) .....	15
Planificación temporal .....	19
Establecimiento del calendario de trabajo .....	20
Identificación de tareas.....	21
Identificación de hitos .....	31
Estimación de la duración de las tareas .....	32
Asignación de recursos .....	36
Dependencias entre tareas.....	37
Diagrama de Gantt .....	42
Conclusiones .....	47
Bibliografía.....	48

# Índice de figuras

Figura 1: Factores de complejidad técnica .....	11
Figura 2: Factores de complejidad del entorno.....	13
Figura 3: EZEstimate - Lista de actores y casos de uso.....	15
Figura 4: EZEstimate - Factores de complejidad técnica.....	16
Figura 5: EZEstimate - Factores de entorno.....	17
Figura 6: EZEstimate - Resultados de la estimación del esfuerzo .....	18
Figura 7: Resumen del Proceso Unificado .....	21
Figura 8: Ejemplo de hito en Windows Project .....	31
Figura 9: Diagrama de Gantt y Planificación Temporal (1) .....	42
Figura 10: Diagrama de Gantt y Planificación Temporal (2) .....	43
Figura 11: Diagrama de Gantt y Planificación Temporal (3) .....	43
Figura 12: Diagrama de Gantt y Planificación Temporal (4) .....	44
Figura 13: Diagrama de Gantt y Planificación Temporal (5) .....	44
Figura 14: Diagrama de Gantt y Planificación Temporal (6) .....	45
Figura 15: Diagrama de Gantt y Planificación Temporal (7) .....	45
Figura 16: Diagrama de Gantt y Planificación Temporal (8) .....	46

# Índice de tablas

Tabla 1: Complejidad de los actores del sistema .....	6
Tabla 2: Complejidad de los casos de uso del sistema .....	9
Tabla 3: Complejidad de los factores de complejidad técnica .....	12
Tabla 4: Complejidad de los factores de complejidad del entorno .....	14
Tabla 5: Descripción de tareas.....	30
Tabla 6: Estimación de las tareas .....	35
Tabla 7: Tabla resumen estimación iteraciones del proyecto .....	35
Tabla 8: Tabla de dependencias entre tareas .....	41

# Introducción

Un plan de proyecto software es un documento en el que se especifican de manera general las actividades, tareas, recursos y plazos necesarios para realizar el desarrollo de un proyecto software. El plan de proyecto estará formado por 2 componentes fundamentales:

- **Estimación del esfuerzo:** Se realizará una estimación del esfuerzo necesario para realizar el desarrollo del proyecto teniendo en cuenta factores como el número y complejidad de actores y casos de uso, así como factores de complejidad técnica y complejidad del entorno. También se tendrá en cuenta los recursos personales.
- **Planificación temporal:** Se realizará una planificación temporal del proyecto a realizar, donde se realizará una descomposición del proyecto en tareas, se asignarán dichas tareas a los miembros del equipo de desarrollo y se establecerán relaciones entre dichas tareas. Esta planificación culminará con la elaboración de un diagrama de Gantt en el que se mostrará toda la planificación temporal del proyecto.

# Estimación del esfuerzo

La estimación del esfuerzo es la primera etapa a realizar dentro del plan del proyecto. En esta etapa, se realizará una estimación o aproximación del esfuerzo necesario para llevar a cabo el desarrollo del proyecto. Para realizar la estimación del esfuerzo, se utilizará el modelo de estimación basado en puntos de casos de uso (UCP) o modelo de Karner.

El modelo de estimación basado en puntos de casos de uso (UCP) se basa en el cálculo de los puntos de casos de uso de un proyecto. Los puntos de casos de uso se calcularán a partir de los siguientes 4 factores:

- ***UUCW (Unadjusted Use Case Weight)***
- ***UAW (Unadjusted Actor Weight)***
- ***TCF (Technical Complexity Factor)***
- ***ECF (Environment Complexity Factor)***

La suma de los 2 primeros factores da como resultado el factor ***UUCP (Unadjusted Use Case Points)***. De esta manera, para obtener los puntos de casos de uso (UCP) se utilizará la siguiente fórmula:

$$UCP = UUCP * TCF * ECF$$

## Complejidad de los actores (UAW)

Para realizar la estimación temporal, en primer lugar, se calculará la complejidad de los distintos actores que van a interactuar con el sistema. Para ello se identificarán los distintos actores que interactúan con el sistema y a cada actor identificado se le asignará una determinada complejidad, siguiendo el criterio que se explicará a continuación.

Tipos de actores según su complejidad:

- **Simple:** Si el actor es un sistema y la aplicación se comunica con él mediante una API.
- **Medio:** Si el actor es un sistema y la aplicación se comunica con él mediante un protocolo (Internet).
- **Complejo:** Persona con una interfaz gráfica.

En el caso particular del sistema a desarrollar, se han identificado 2 actores que interactúan con el sistema.

Por una parte, existe un **actor usuario** el cual se trata de una persona con una interfaz gráfica por lo que se trata de un **actor complejo**.

Por otra parte, se tiene un **actor sistema** el cual se trata de un sistema donde la aplicación se comunica con él mediante una API, por lo que se trata de un **actor simple**.

Actor	Complejidad
<i>Usuario</i>	<i>Complejo</i>
<i>Sistema</i>	<i>Simple</i>

Tabla 1: Complejidad de los actores del sistema

## Complejidad de los casos de uso (UUCW)

Una vez se ha determinado la complejidad de los distintos actores que van a interactuar con el sistema, ahora se va a calcular la complejidad de los distintos casos de uso que forman parte del sistema. Para determinar la complejidad de los casos de uso, se determinará el número de transacciones que tenga cada caso de uso y en función de ese número de transacciones se asignará una complejidad a ese caso de uso.

Para ello, es muy importante tener en cuenta el concepto de transacción para poder contar las transacciones de cada caso de uso correctamente y poder asignar a cada caso de uso la complejidad “correcta”.

Cada acción individual realizada por el usuario o el sistema no se considera una transacción, es decir, una **acción** no equivale a una **transacción**.

Una **transacción** es una o varias acciones realizadas por el usuario seguida de una o varias acciones realizadas por el sistema en respuesta a la acción realizada por el usuario.

Una vez se tenga el número de transacciones que tenga cada caso de uso, se asignará una determinada complejidad a cada caso de uso identificado, teniendo en cuenta los siguientes criterios:

- Si un caso de uso tiene menos de 4 transacciones, la complejidad de ese caso de uso será **simple**.
- Si un caso de uso tiene entre 4 y 7 transacciones, la complejidad de ese caso de uso será **medio**.
- Si un caso de uso tiene más de 7 transacciones, la complejidad de ese caso de uso será **complejo**.

Una vez se ha introducido el concepto de transacción, se realizará el conteo del número de transacciones que existe en cada caso de uso y se asignará una complejidad a cada caso de uso en función del número de transacciones.

Hay que tener en cuenta, las excepciones para realizar el conteo de las transacciones de un caso de uso, ya que, aunque no forman parte de la secuencia normal de pasos, es un camino de ejecución que se puede tomar y que debe ser correctamente implementado.



<b>Caso de uso</b>	<b>Número de transacciones</b>	<b>Complejidad</b>
<b>UC-01 Consultar información sobre la configuración del protocolo</b>	1	Simple
<b>UC-02 Modificar información sobre la configuración del protocolo</b>	3	Simple
<b>UC-03 Consultar información sobre el modo de trabajo de la estación</b>	1	Simple
<b>UC-04 Modificar información sobre el modo de trabajo de la estación</b>	3	Simple
<b>UC-05 Consultar información sobre la configuración del canal</b>	1	Simple
<b>UC-06 Modificar información sobre la configuración del canal</b>	3	Simple
<b>UC-07 Establecer conexión física</b>	1	Simple
<b>UC-08 Finalizar el establecimiento de la conexión física</b>	1	Simple
<b>UC-09 Enviar trama de información</b>	6	Media
<b>UC-10 Enviar trama de receptor preparado (RR)</b>	4	Media
<b>UC-11 Enviar trama de receptor no preparado (RNR)</b>	4	Media
<b>UC-12 Enviar trama de rechazo (REJ)</b>	4	Media
<b>UC-13 Enviar trama de rechazo selectivo (SREJ)</b>	4	Media
<b>UC-14 Enviar trama de petición de conexión (SABM)</b>	4	Media
<b>UC-15 Enviar trama de petición de desconexión (DISC)</b>	4	Media
<b>UC-16 Enviar trama de asentimiento no numerado (UA)</b>	6	Media
<b>UC-17 Enviar trama de modo desconectado (DM)</b>	6	Media
<b>UC-18 Enviar trama de rechazo de trama (FRMR)</b>	4	Media
<b>UC-19 Representar gráficamente el envío de una trama</b>	1	Simple
<b>UC-20 Ver detalle de la trama</b>	1	Simple
<b>UC-21 Guardar captura de tráfico</b>	3	Simple
<b>UC-22 Cargar captura de tráfico</b>	5	Media
<b>UC-23 Implementar timeout ante COMMAND</b>	1	Simple
<b>UC-24 Implementar timeout ante trama I</b>	1	Simple
<b>UC-25 Implementar timeout ante REQUEST</b>	1	Simple
<b>UC-26 Recibir trama de información</b>	1	Simple
<b>UC-27 Recibir trama de receptor preparado (RR)</b>	1	Simple

<b>UC-28 Recibir trama de receptor no preparado (RNR)</b>	1	Simple
<b>UC-29 Recibir trama de rechazo (REJ)</b>	1	Simple
<b>UC-30 Recibir trama de rechazo selectivo (SREJ)</b>	1	Simple
<b>UC-31 Recibir trama de petición de conexión (SABM)</b>	1	Simple
<b>UC-32 Recibir trama de petición de desconexión (DISC)</b>	1	Simple
<b>UC-33 Recibir trama de asentimiento no numerado (UA)</b>	1	Simple
<b>UC-34 Recibir trama de modo desconectado (DM)</b>	1	Simple
<b>UC-35 Recibir trama de rechazo de trama (FRMR)</b>	1	Simple
<b>UC-36 Recibir trama errónea</b>	1	Simple
<b>UC-37 Representar gráficamente la recepción de una trama</b>	1	Simple

Tabla 2: Complejidad de los casos de uso del sistema

## Factores de complejidad técnica (TCF)

El modelo de puntos de casos de uso define **13 factores de complejidad técnica** de forma que a cada factor se le asigna un **peso** de acuerdo con su impacto en el proyecto y una **complejidad percibida** en función a la complejidad percibida por el equipo de desarrollo.

El **peso** de cada factor de complejidad se encuentra asignado de manera predefinida por el modelo de puntos de casos de uso. La **complejidad percibida** de cada factor de complejidad será un número entero entre 0 y 5.

Los 13 factores de complejidad técnica son los siguientes:

- **Sistemas distribuidos:** El sistema se distribuye en distintos nodos, lo que supone una complejidad técnica adicional.
- **Rendimiento:** El sistema debe proporcionar las distintas funcionalidades siguiendo unos criterios de tiempo de respuesta, lo que supone una complejidad técnica adicional.
- **Eficiencia del usuario final:** El usuario debe poder realizar las tareas deseadas con la mayor eficiencia posible, lo que supone una complejidad técnica adicional.
- **Procesamiento interno complejo:** El sistema tiene que realizar una serie de cálculos internos para proporcionar las distintas funcionalidades, lo que supone una complejidad técnica adicional.
- **Reusabilidad:** Se busca reutilizar ciertas porciones de código para aumentar la eficiencia y calidad del sistema.
- **Facilidad de instalación:** La implementación de una instalación sencilla puede implicar una complejidad técnica adicional.
- **Facilidad de uso:** Diseñar un sistema fácil de usar para el usuario puede implicar una complejidad técnica adicional.
- **Portabilidad:** Diseñar un sistema que funcione en distintas plataformas puede implicar una complejidad técnica adicional.
- **Facilidad de cambio:** Diseñar un sistema en el que se puedan realizar cambios de manera rápida y sencilla puede implicar una complejidad técnica adicional.
- **Concurrencia:** Un sistema que maneja múltiples procesos y/o usuarios de manera concurrente supone una complejidad técnica adicional.
- **Características especiales de seguridad:** Diseñar un sistema seguro contra posibles ataques puede implicar una complejidad técnica adicional.
- **Accesos directo a terceras partes:** Un sistema que permita o proporcione acceso a otros programas ajenos de terceros, puede implicar una complejidad técnica adicional.
- **Se requiere entrenamiento especial del usuario:** Un sistema que necesite instruir a los usuarios para que estos puedan hacer uso del mismo, puede implicar una complejidad técnica adicional.

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 1: Factores de complejidad técnica

Factor de complejidad técnica	Peso	Complejidad percibida	Justificación
<b>Sistemas distribuidos</b>	2	0	El sistema a desarrollar será una aplicación de escritorio local, por lo que no habrá ningún tipo de comunicación con otros sistemas.
<b>Rendimiento</b>	1	2	El sistema debe realizar el intercambio de tramas, así como su representación de una manera eficiente y rápida, pero no se trata de un requisito crítico.
<b>Eficiencia del usuario final</b>	1	4	El sistema trata de una temática bastante específica como es la simulación del protocolo HDLC. Debido a esto, se necesita de un gran esfuerzo para lograr una buena eficiencia en los usuarios finales.
<b>Procesamiento interno complejo</b>	1	3	El sistema debe encargarse de generar y recibir las tramas de manera correcta, así como gestionar <i>timeouts</i> entre otras tareas. No se trata de una mera aplicación de almacenamiento y procesamiento simple de datos.
<b>Reusabilidad</b>	1	3	El sistema implementará muchas funcionalidades similares que requieren de uso de ciertas porciones de código compartidas
<b>Facilidad de instalación</b>	0.5	1	La instalación del sistema consta de la ejecución de un ejecutable. Por tanto, el sistema es bastante fácil de instalar.
<b>Facilidad de uso</b>	0.5	4	El desarrollo del sistema se centrará entre otras cosas, en la creación de una interfaz de usuario fácil de usar. Se utilizarán distintas metodologías y técnicas para lograr este cometido.
<b>Portabilidad</b>	2	1	El sistema está destinado a funcionar en el sistema operativo Windows.
<b>Facilidad de cambio</b>	1	3	El sistema en el futuro puede evolucionar e incluir nuevos modos de funcionamiento del protocolo HDLC, por lo que es conveniente diseñar un sistema que facilite los distintos cambios que se puedan introducir.
<b>Concurrencia</b>	1	3	El sistema estará formado por 2 procesos principales conectados a través de una tubería. Se tendrán distintos hilos para realizar ciertas labores sencillas del protocolo HDLC. Sin embargo, solo existe un único usuario en el sistema.

<b><i>Características especiales de seguridad</i></b>	1	1	El sistema a desarrollar se trata de un simulador local del protocolo HDLC por lo que no hay información relevante que proteger.
<b><i>Acceso directo a terceras partes</i></b>	1	0	El sistema no proporcionará servicios a otros programas ajenos
<b><i>Se requiere entrenamiento especial del usuario</i></b>	1	3	Aunque no es obligatorio, en ciertos casos será necesario realizar una instrucción a los usuarios debido a la complejidad de la temática del sistema. Se elaborarán distintos mecanismos de ayuda que permitan al usuario encender el funcionamiento del protocolo y de la aplicación.

Tabla 3: Complejidad de los factores de complejidad técnica

Como se puede observar, los factores de complejidad técnica suponen un incremento en el esfuerzo del desarrollo del proyecto. Es por esto, por lo que los factores de complejidad técnica tienen valores positivos ya que incrementaron el resultado final del esfuerzo estimado.

## Factores de complejidad del entorno (ECF)

El modelo de puntos de casos de uso define **8 factores de complejidad del entorno** de forma que a cada factor se le asigna un **peso** de acuerdo con su impacto en el proyecto y una **complejidad percibida** en función a la complejidad percibida por el equipo de desarrollo.

El **peso** de cada factor de complejidad se encuentra asignado de manera predefinida por el modelo de puntos de casos de uso. La **complejidad percibida** de cada factor del entorno será un número entero entre 0 y 5.

Los 8 factores de complejidad del entorno son los siguientes:

- **Familiaridad con UML:** La experiencia del equipo de desarrollo con el entorno de UML puede implicar un aumento de la eficiencia del equipo y una disminución de la complejidad técnica.
- **Trabajadores a tiempo parcial:** Distribuir las tareas entre distintos trabajadores puede implicar un esfuerzo adicional en la coordinación.
- **Capacidad de los analistas:** El conocimiento experto de los analistas puede influir positivamente en el desarrollo del proyecto.
- **Experiencia en la aplicación:** La experiencia en aplicaciones similares a la aplicación a desarrollar puede facilitar el desarrollo del sistema, así como la comprensión del mismo.
- **Experiencia en orientación a objetos:** La experiencia del equipo de desarrollo en el paradigma orientado a objetos puede facilitar el desarrollo de sistema, así como la calidad del mismo.
- **Motivación:** La motivación del equipo de desarrollo puede influir en el desarrollo del proyecto.
- **Dificultad del lenguaje de programación:** La dificultad en el lenguaje de programación utilizado en el desarrollo puede suponer un aumento de la complejidad técnica.
- **Estabilidad de los requisitos:** La estabilidad de los requisitos implica un menor esfuerzo en el desarrollo debido al número reducido de cambios a realizar en el transcurso del desarrollo del proyecto.

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 2: Factores de complejidad del entorno

<b>Factor de complejidad del entorno</b>	<b>Peso</b>	<b>Complejidad percibida</b>	<b>Justificación</b>
<b><i>Familiaridad con UML</i></b>	1.5	2	A lo largo del grado, se ha manejado en varias ocasiones el lenguaje UML. Sin embargo, no se considera que el conocimiento sobre UML sea elevado.
<b><i>Trabajadores a tiempo parcial</i></b>	-1	1	El equipo de desarrollo está formado por un único integrante, por lo que no será necesaria la coordinación entre miembros del equipo.
<b><i>Capacidad de los analistas</i></b>	0.5	2	Se tiene cierta capacidad de análisis y conocimiento experto sobre el protocolo HDLC, pero dicha capacidad de análisis no es grande, debido a la falta de experiencia del desarrollador en proyectos software de envergadura.
<b><i>Experiencia en la aplicación</i></b>	0.5	4	Existe bastante familiaridad con el sistema a desarrollar, ya que se ha utilizado en gran medida otro simulador del protocolo HDLC en la asignatura de Redes de computadores I del grado.
<b><i>Experiencia en orientación a objetos</i></b>	1	3	A lo largo del grado, se han utilizado varios lenguajes orientados a objetos como Java o C#. Sin embargo, no se considera que la experiencia en la orientación a objetos sea muy elevada.
<b><i>Motivación</i></b>	1	5	Dado a que se trata de un proyecto de final de grado cuya temática ha sido elegida por el propio desarrollador, la motivación para llevar a cabo el proyecto es máxima.
<b><i>Dificultad del lenguaje de programación</i></b>	-1	1	El lenguaje C# se considera ampliamente como un lenguaje de dificultad media-baja, el cual dispone de muchos recursos y documentación. El entorno de WPF (C# y XAML) ha sido previamente utilizado por el desarrollador, lo que disminuye la percepción de dificultad del lenguaje de programación.
<b><i>Estabilidad en los requisitos</i></b>	2	3	Los requisitos del sistema a desarrollar parecen estables en primera instancia. No se prevén grandes cambios en los requisitos, pero es posible que surjan durante el desarrollo del proyecto nuevas funcionalidades a incluir en el proyecto.

Tabla 4: Complejidad de los factores de complejidad del entorno

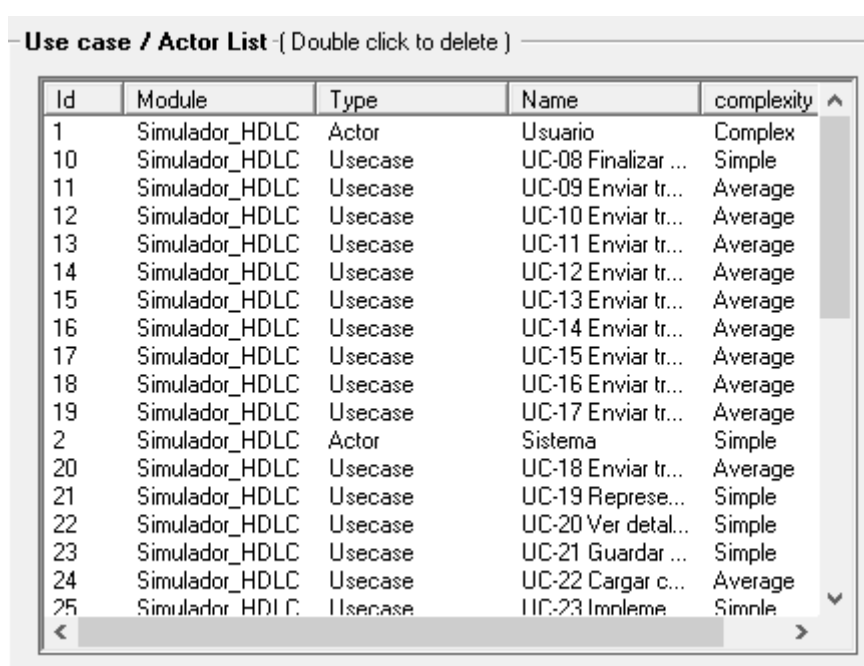
Como se puede observar, los factores de complejidad del entorno por lo general suponen una disminución en el esfuerzo del desarrollo del proyecto. Debido a esto, para calcular el ECF se multiplicará el valor de cada factor por un factor negativo. Los trabajadores a tiempo parcial y la dificultad del lenguaje de programación son factores que incrementan el esfuerzo de desarrollo. Es por esto por lo que estos 2 factores tienen un peso negativo, ya que, multiplicados por el factor negativo, dan como resultado un número positivo que suma esfuerzo al proyecto.

## Cálculo de la estimación del esfuerzo (*EZEstimate*)

Una vez se ha establecido la complejidad de los actores, casos de uso, así como los valores de complejidad a los factores de complejidad técnica y complejidad del entorno, el último paso a dar en la estimación del esfuerzo es obtener el cálculo de los puntos de casos de uso (UCP) a partir de la información existente y realizar la conversión a horas de persona.

Para realizar el cálculo de los puntos de casos de uso (UCP), se utilizará la herramienta *EZEstimate*, la cual, permite estimar el esfuerzo de un proyecto a partir de los puntos de casos de uso.

En primer lugar, se introducirá en *EZEstimate* los datos relativos a los **actores** y los **casos de uso** indicando para cada actor/caso de uso el nombre y su complejidad asignada (Simple, Media o Compleja). En el caso particular de este proyecto, la lista de actores y casos de uso tendrá el siguiente aspecto:



Id	Module	Type	Name	complexity
1	Simulador_HDLC	Actor	Usuario	Complex
10	Simulador_HDLC	Usecase	UC-08 Finalizar ...	Simple
11	Simulador_HDLC	Usecase	UC-09 Enviar tr...	Average
12	Simulador_HDLC	Usecase	UC-10 Enviar tr...	Average
13	Simulador_HDLC	Usecase	UC-11 Enviar tr...	Average
14	Simulador_HDLC	Usecase	UC-12 Enviar tr...	Average
15	Simulador_HDLC	Usecase	UC-13 Enviar tr...	Average
16	Simulador_HDLC	Usecase	UC-14 Enviar tr...	Average
17	Simulador_HDLC	Usecase	UC-15 Enviar tr...	Average
18	Simulador_HDLC	Usecase	UC-16 Enviar tr...	Average
19	Simulador_HDLC	Usecase	UC-17 Enviar tr...	Average
2	Simulador_HDLC	Actor	Sistema	Simple
20	Simulador_HDLC	Usecase	UC-18 Enviar tr...	Average
21	Simulador_HDLC	Usecase	UC-19 Represe...	Simple
22	Simulador_HDLC	Usecase	UC-20 Ver detal...	Simple
23	Simulador_HDLC	Usecase	UC-21 Guardar ...	Simple
24	Simulador_HDLC	Usecase	UC-22 Cargar c...	Average
25	Simulador_HDLC	Usecase	UC-23 Impleme...	Simple

Figura 3: *EZEstimate* - Lista de actores y casos de uso

En segundo lugar, se introducirá en *EZEstimate* los datos relativos a los **factores de complejidad técnica** indicando para cada factor de complejidad técnica su complejidad asignada (valores del 0 al 5). En el caso particular de este proyecto, los valores asignados a los factores de complejidad técnica serán los siguientes:



Factor	Relevance
Distributed system	0
Response / Throughput performance objectives	2
End-user efficiency	4
Complex internal processing	3
Reusable code	3
Easy to install	1
Easy to use	4
Portable	1
Easy to change	3
Concurrent	3
Includes security features	1
Third party access	0
Special user training facilities required	3

Figura 4: *EZEstimate* - Factores de complejidad técnica

En tercer lugar, se introducirá en *EZEstimate* los datos relativos a los **factores de entorno** indicando para cada factor de complejidad del entorno su complejidad asignada (valores del 0 al 5). En el caso particular de este proyecto, los valores asignados a los factores de complejidad del entorno serán los siguientes:

Factor	Relevance
Familiar with Rational unified process	2
Application experience	4
Object oriented experience	3
Lead analyst capability	2
Motivation	5
Stable requirements	3
Part-time workers	1
Difficult programming language	1

OK Cancel

Figura 5: *EZEstimate* - Factores de entorno

Con estos datos introducidos, la herramienta *EZEstimate* será capaz de calcular los puntos de casos de uso (UCP). Para ello, calculará *UUCW*, *UAW*, *TCF* y *ECF* a partir de los datos introducidos y los pesos que maneja por defecto la herramienta.

Para convertir los puntos de casos de uso a horas de persona, se utilizará un factor de conversión el cual puede ser modificado por el usuario en *EZEstimate*. En este proyecto, vamos a suponer un factor de conversión de 6 horas por cada punto de caso de uso.

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

*Factor de conversión (número de horas de persona por UCP)*

Teniendo en cuenta este factor de conversión, los resultados obtenidos en la estimación del esfuerzo utilizando la herramienta *EZEstimate* son los siguientes:

Estimation Summary	
UAW	4
UUCW	240
UUPC = UAW + UUCW	244
TFactor	28
EFactor	18
TCF = 0.6 + (.01*TFactor)	0.88
EF = 1.4 + (-0.03*EFactor)	0.86
UCP = UUCP*TCT*EF	184,6592
<b>Total Effort@</b> 6 Hrs/UCP	1107,9552

Figura 6: *EZEstimate* - Resultados de la estimación del esfuerzo

Como se puede observar, el resultado final es de 1107 horas de persona, las cuales se repartirán de manera exclusiva al único miembro del equipo de desarrollo.

# Planificación temporal

La estimación de tareas consiste en la identificación de tareas, la asignación de tiempos y recursos a estas tareas y la secuencia de ejecución de las mismas (dependencias entre tareas) con el objetivo de reducir el tiempo de desarrollo del proyecto. De esta manera, el esfuerzo estimado anteriormente se va a distribuir entre las distintas tareas del proyecto.

Para realizar la planificación temporal, vamos a seguir una serie de pasos. Los pasos a seguir para realizar la planificación temporal es el siguiente:

1. **Establecimiento del calendario de trabajo:** En este primer paso, se define el horario de trabajo que vamos a seguir los miembros del equipo de desarrollo. En este calendario se incluirán festivos, fines de semana y acontecimientos especiales que difieran del horario de trabajo normal.
2. **Identificación de tareas:** En este segundo paso, se descompone el proyecto en un conjunto de tareas y estas tareas se distribuirán en distintas iteraciones y distintas fases siguiendo el proceso unificado.
3. **Identificación de hitos:** En este tercer paso, se definen los puntos en los que finaliza un conjunto de tareas pertenecientes a una iteración y comienzan las tareas de otra iteración.
4. **Asignación de tiempos:** En este cuarto paso, se define el tiempo o duración de cada una de las tareas que conforman el proyecto.
5. **Asignación de recursos a las tareas:** En este quinto paso, se asignan los distintos recursos personales a las diferentes tareas identificadas. Con esto nos aseguramos que tenemos los recursos necesarios para realizar las distintas tareas.
6. **Dependencias entre tareas:** En este sexto paso, se establecen las dependencias o relaciones entre las tareas que pertenezcan a la misma iteración. Estas dependencias entre tareas supone que unas tareas deben esperar a que se finalizan las tareas que las preceden y esto afecta a la duración del proyecto.

A continuación, se explicará de manera detallada cómo se han realizado cada uno de estos pasos, para obtener así la planificación temporal del proyecto.

## Establecimiento del calendario de trabajo

El primer paso a realizar para la planificación temporal del proyecto es establecer el calendario de trabajo de los miembros del equipo de desarrollo. En el caso particular de este proyecto, hay que tener en cuenta que el equipo de desarrollo está formado por un único miembro.

Como el miembro del equipo de desarrollo se trata de un estudiante del grado de Ingeniería Informática, se diseñará una jornada de trabajo estándar ya que compaginar el desarrollo del proyecto con otras labores (clases, etc.) puede resultar ciertamente complicado.

De esta manera, el diseño del calendario de trabajo va a constar de jornadas de trabajo estándar de 8 horas. Estas jornadas de trabajo pueden transcurrir en cualquier momento del día, dependiendo de la situación particular del miembro de equipo, por lo que el horario será flexible en ese aspecto siempre y cuando se realicen las 8 horas de trabajo.

Los fines de semana (sábados y domingos) se considerarán como días no laborables no por lo que no se tendría en cuenta esos días para la planificación temporal.

Teniendo en cuenta el diseño de las jornadas, se introduce esta información en la herramienta Windows Project, para que nos elabore automáticamente el calendario de trabajo a partir de esta información y la información sobre la fecha de comienzo del desarrollo del proyecto.

Por otra parte, se establecerá como fecha de comienzo del proyecto el día 22 de diciembre de 2023. Se ha elegido esta fecha como día de comienzo ya que coincide con el fin del primer cuatrimestre y a partir de esta fecha, el desarrollador puede centrarse en mayor medida en el desarrollo del proyecto.

## Identificación de tareas

Una vez se ha establecido correctamente el horario de trabajo que deben seguir los miembros del equipo de desarrollo, el siguiente paso es descomponer el proyecto en un conjunto de tareas. Para realizar esta descomposición de tareas, se van a seguir las directrices del Proceso Unificado.

En el Proceso Unificado el ciclo de vida del software se dividía en 4 fases. Estas 4 fases son las siguientes:

- **Fase de inicio:** En esta fase se desarrolla una descripción del producto final, y se presenta el análisis del negocio.
- **Fase de elaboración:** En esta fase se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.
- **Fase de construcción:** En esta fase se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo.
- **Fase de transición:** En esta fase el producto se convierte en la versión beta (se realiza la transición a los usuarios).

Cada fase se subdivide en iteraciones. El número de iteraciones varía en función de la fase a la que pertenecen. En cada iteración se desarrolla en secuencia un conjunto de disciplinas.

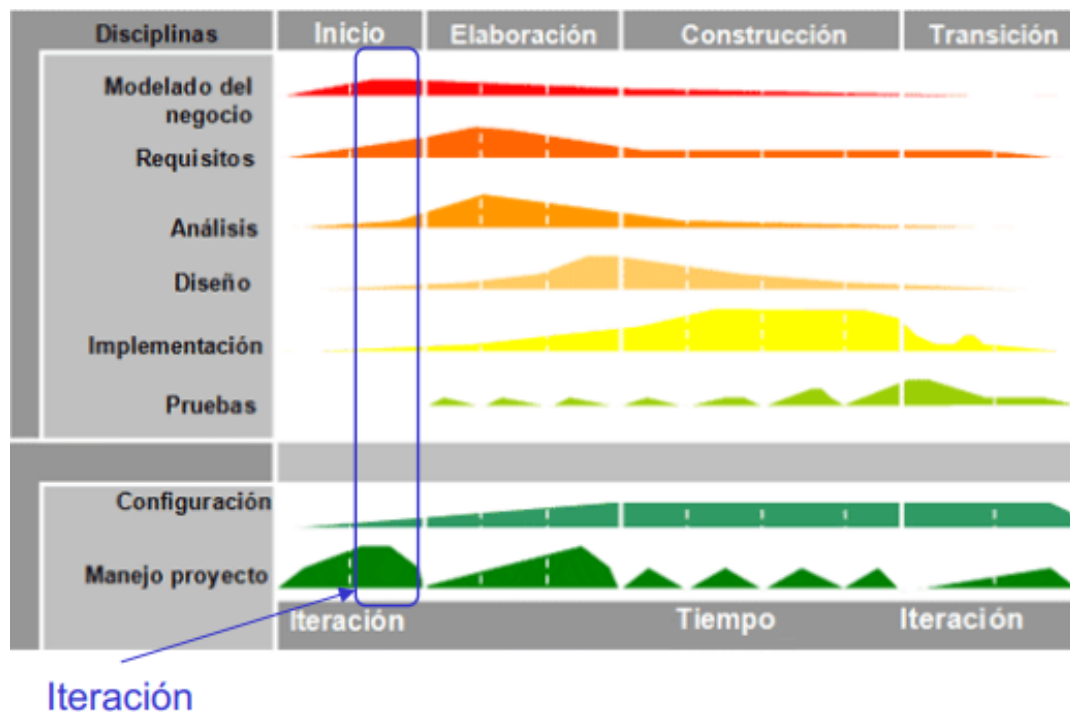


Figura 7: Resumen del Proceso Unificado

Cada disciplina es un conjunto de actividades relacionadas vinculadas a un área específica dentro del proyecto total. Las más importantes son: Modelado de Negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas.

De esta manera, el objetivo es identificar los conjuntos de actividades o tareas en los que se descompone el desarrollo de un proyecto y los cuales se agrupan en disciplinas e iteraciones.

Dentro de cada iteración, se definirán un subconjunto de las tareas de desarrollo del proyecto y agruparán dichas tareas en las disciplinas de Modelado de Negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas. De esta manera, a lo largo del desarrollo del proyecto en cada iteración se irán realizando tareas de todas disciplinas con una mayor o menor carga de trabajo.

En el caso particular de este proyecto, se dividirá el desarrollo del proyecto en 8 iteraciones donde la distribución las tareas identificadas para el desarrollo de este proyecto será la siguiente:

Nombre		Descripción
0	<b>Inicio del proyecto</b>	Marca temporal que representa el comienzo del desarrollo del proyecto.
1	<b>Inicio</b>	Marca temporal que representa el comienzo de la fase de "Inicio".
2	<b>Iteración 1</b>	Marca temporal que representa el comienzo de la iteración 1 de la fase de "Inicio".
2.1	<b>Modelado de negocio</b>	
2.1.1	<b>Investigar sobre el funcionamiento teórico del protocolo HDLC</b>	Se realizará un repaso de los conceptos básicos del funcionamiento del protocolo HDLC, y en concreto del modo de operación balanceado asíncrono (ABM).
2.1.2	<b>Investigar proyectos similares existentes</b>	Se realizará una investigación de proyectos que traten una temática similar al proyecto que se desea desarrollar.
2.2	<b>Requisitos</b>	
2.2.1	<b>Reuniones con el cliente</b>	Se realizarán reuniones con los clientes (en este caso, los tutores del proyecto) para obtener una idea general de cómo debe ser el sistema a desarrollar.
2.2.2	<b>Identificación y definición de los primeros objetivos</b>	Se identificarán los objetivos principales que se deben perseguir en el desarrollo del proyecto.
2.2.3	<b>Determinar el ámbito del sistema</b>	Se determinará el público objetivo al que va dirigido el desarrollo del sistema.
2.3	<b>Análisis</b>	
2.3.1	<b>Realización de un primer boceto de la arquitectura</b>	Se realizará una primera aproximación inicial sobre la posible arquitectura del sistema.
2.4	<b>Diseño</b>	
2.4.1	<b>Distribución de los subsistemas</b>	Se distribuye la funcionalidad del sistema a desarrollar en distintos subsistemas. Estos subsistemas serán conceptuales.
2.5	<b>Implementación</b>	
2.5.1	<b>Descarga e instalación del</b>	Se preparará el entorno de desarrollo en el cual

		<b>entorno de desarrollo</b>	se implementará posteriormente el sistema.
	2.6	<b>Pruebas</b>	
	2.6.1	<b>Pruebas sobre el correcto funcionamiento del entorno de desarrollo</b>	Se comprobará el correcto funcionamiento del entorno de desarrollo, en el cual se implementará posteriormente el sistema.
3	<b>Hito fin iteración 1</b>		Este hito representa el fin de la primera iteración de la fase de "Inicio".
4	<b>Hito fin fase "Inicio"</b>		Este hito representa el fin de la fase de "Inicio".
5	<b>Elaboración</b>		Marca temporal que representa el comienzo de la fase de "Elaboración".
6	<b>Iteración 1</b>		Marca temporal que representa el comienzo de la iteración 1 de la fase de "Elaboración".
	6.1	<b>Modelado de negocio</b>	
	6.1.1	<b>Investigación sobre técnicas de comunicación entre procesos</b>	Se realizará una investigación sobre distintos mecanismos para la comunicación entre procesos, evaluando las ventajas e inconvenientes de cada uno de ellos.
	6.2	<b>Requisitos</b>	
	6.2.1	<b>Identificación de actores</b>	Se identificarán los distintos actores (roles) que interactúan con el sistema.
	6.2.2	<b>Identificar requisitos no funcionales</b>	Se identificarán los requisitos no funcionales del proyecto a desarrollar, teniendo en cuenta que son requisitos críticos que debe cumplir el sistema.
	6.2.3	<b>Identificar requisitos de información</b>	Se identificarán las necesidades básicas de almacenamiento de información dentro del proyecto.
	6.2.4	<b>Identificación de los paquetes funcionales</b>	Se identificarán los distintos paquetes funcionales en los cuales se distribuirán las distintas funcionalidades del sistema.
	6.3	<b>Análisis</b>	
	6.3.1	<b>Identificación de las primeras clases entidad</b>	Se identificarán las clases entidad más importantes, las cuales formarán parte del diagrama de clases.
	6.4	<b>Diseño</b>	
	6.4.1	<b>Definición general del mapa del sitio</b>	Se describirán de manera genérica las distintas pantallas e interfaces gráficas que incuria la aplicación a desarrollar.
	6.5	<b>Implementación</b>	
	6.5.1	<b>Elaboración de un primer prototipo</b>	Se creará un pequeño boceto que contenga la representación visual básica del sistema a desarrollar.
	6.6	<b>Pruebas</b>	
	6.6.1	<b>Realizar pruebas en entornos similares</b>	Se realizarán pruebas en entornos de simulación similares para definir de manera más clara cómo se desea desarrollar el sistema.



7	<b>Hito fin iteración 1</b>		Este hito representa el fin de la primera iteración de la fase de “Elaboración”.
8	<b>Iteración 2</b>		Marca temporal que representa el comienzo de la iteración 2 de la fase de “Elaboración”.
	8.1	<b>Modelado de negocio</b>	
	8.1.1	<b>Investigación sobre la creación de hilos en el lenguaje de programación utilizado</b>	Se investigará sobre la creación y manejo de hilos en el lenguaje de programación utilizado, para aprender a gestionar las operaciones concurrentes.
	8.2	<b>Requisitos</b>	
	8.2.1	<b>Identificación de casos de uso relacionados con la gestión de la configuración del simulador</b>	Se identificarán y definirán los casos de uso relacionados con la configuración del simulador.
	8.2.2	<b>Identificación de casos de uso relacionados con el envío de tramas</b>	Se identificarán y definirán los casos de uso relacionados con el envío de tramas dentro del simulador.
	8.2.3	<b>Realización de encuestas de contenido a los usuarios</b>	Se obtendrá cierta información de los usuarios sobre las funcionalidades que esperan del simulador, así como la interacción con el sistema.
	8.3	<b>Análisis</b>	
	8.3.1	<b>Identificación del resto de clases entidad y las relaciones entre ellas</b>	Se identificarán el resto de las clases entidad, las cuales formarán parte del diagrama de clases, así como las relaciones entre dichas clases entidad.
	8.3.2	<b>Identificación de los primeros paquetes</b>	Se identificarán los primeros paquetes, teniendo en cuenta los paquetes funcionales identificados previamente en la disciplina de Requisitos.
	8.4	<b>Diseño</b>	
	8.4.1	<b>Evaluación de decisiones de diseño de la interfaz gráfica</b>	Se tomarán distintas decisiones de diseño sobre la interfaz gráfica del sistema a desarrollar como el tipo de fuente o la elección de colores.
	8.5	<b>Implementación</b>	
	8.5.1	<b>Implementación de un modelo de comunicación entre 2 procesos</b>	Se implementará una aplicación básica que permita intercambiar un mensaje de un proceso a otro.
	8.6	<b>Pruebas</b>	
	8.6.1	<b>Definición de casos de prueba</b>	Se definirán posibles casos de prueba, que serán utilizados posteriormente para evaluar el funcionamiento del sistema desarrollado.
9	<b>Hito fin iteración 2</b>		Este hito representa el fin de la segunda iteración de la fase de “Elaboración”.
10	<b>Hito fin fase “Elaboración”</b>		Este hito representa el fin de la fase de “Elaboración”.
11	<b>Construcción</b>		Marca temporal que representa el comienzo de la fase de “Construcción”.

12	<b>Iteración 1</b>		Marca temporal que representa el comienzo de la iteración 1 de la fase de "Construcción".
	12.1	<b>Modelado de negocio</b>	
		12.1.1	<b>Identificación de riesgos</b> Se identificarán los posibles riesgos que puedan comprometer el desarrollo del proyecto.
	12.2	<b>Requisitos</b>	
		12.2.1	<b>Identificación de casos de uso relacionados con la gestión de las capturas de tráfico</b> Se identificarán y definirán los casos de uso relacionados con el guardado y cargado de capturas de tráfico en el simulador.
		12.2.2	<b>Identificación de casos de uso relacionados con la aplicación de <i>timeouts</i></b> Se identificarán y definirán los casos de uso relacionados con la aplicación de los distintos tipos de <i>timeouts</i> dentro del simulador.
		12.2.3	<b>Identificación de casos de uso relacionados con la recepción de tramas</b> Se identificarán y definirán los casos de uso relacionados con la recepción de tramas dentro del simulador.
	12.3	<b>Análisis</b>	
		12.3.1	<b>Identificación de clases de control y clases interfaz</b> Se identificarán todas las clases de control y las clases interfaz del sistema a desarrollar.
		12.3.2	<b>Identificación del resto de paquetes</b> Se identificarán los paquetes restantes, teniendo en cuenta los paquetes funcionales identificados previamente en la disciplina de Requisitos.
		12.3.3	<b>Realización de los casos de uso relacionados con la gestión de la configuración del simulador</b> Se llevará a cabo la realización a nivel de análisis de los casos de uso relacionados con la gestión de la configuración del simulador.
		12.3.4	<b>Realización de los casos de uso relacionados con el envío de tramas</b> Se llevará a cabo la realización a nivel de análisis de los casos de uso relacionados con la envío de tramas dentro del simulador.
	12.4	<b>Diseño</b>	
		12.4.1	<b>Identificación de los primeros subsistemas</b> Se identificarán los primeros subsistemas, teniendo en cuenta los paquetes del análisis identificados previamente en la disciplina de Análisis.
		12.4.2	<b>Elaboración de un esquema de la visualización de la interfaz gráfica</b> Se elaborará un esquema o <i>wireframe</i> de las pantallas de la interfaz previamente identificadas en las que se realizará una distribución de los elementos visuales básicos.
		12.4.3	<b>Identificación las primeras clases de diseño</b> Se identificarán las primeras clases de diseño, las cuales formarán parte del diagrama de clases diseño.
	12.5	<b>Implementación</b>	
		12.5.1	<b>Implementación de hilos que realicen tareas simples y recursivas</b> Se implementará una aplicación básica que creará hilos los cuales realizarán tareas simples y manera recursiva.
	12.6	<b>Pruebas</b>	
		12.6.1	<b>Definir una estrategia para la realización de pruebas</b> Se elaborará un plan para ejecutar los distintos casos de prueba identificados previamente.

13	Hito fin iteración 1		Este hito representa el fin de la primera iteración de la fase de “Construcción”.	
14	Iteración 2		Marca temporal que representa el comienzo de la iteración 2 de la fase de “Construcción”.	
	14.1	Modelado de negocio		
		14.1.1	Priorización de riesgos	Se ordenarán los riesgos identificados previamente en función de la importancia, el impacto y la probabilidad del riesgo.
	14.2	Requisitos		
		14.2.1	Identificación de escenarios de uso	Se identificarán los distintos escenarios de uso en los cuales se puede encontrar el usuario en su interacción con el sistema.
		14.2.2	Elaboración del diagrama de casos de uso	Se establecerán relaciones entre los distintos casos de uso y actores identificados previamente y se distribuirán en función de los paquetes funcionales previamente identificados.
	14.3	Análisis		
		14.3.1	Realización de los casos de uso relacionados con la gestión de las capturas de tráfico	Se llevará a cabo la realización a nivel de análisis de los casos de uso relacionados con el guardado y cargado de capturas de tráfico en el simulador.
		14.3.2	Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i>	Se llevará a cabo la realización a nivel de análisis de los casos de uso relacionados con la aplicación de los distintos tipos de <i>timeouts</i> dentro del simulador.
		14.3.3	Realización de los casos de uso relacionados con la recepción de tramas	Se llevará a cabo la realización a nivel de análisis de los casos de uso relacionados con la recepción de tramas dentro del simulador.
		14.3.4	Análisis de tareas	Se realizará el análisis de las distintas tareas u operaciones que puede realizar el usuario dentro del sistema.
	14.4	Diseño		
		14.4.1	Identificación del resto de clases de diseño	Se identificarán el resto de las clases de diseño, las cuales formarán parte del diagrama de clases de diseño.
14.4.2		Identificación de resto de subsistemas de diseño	Se identificarán los subsistemas restantes, teniendo en cuenta los paquetes del análisis identificados previamente en la disciplina de Análisis.	
14.4.3		Realización de los casos de uso relacionados con la gestión de la configuración del simulador	Se llevará a cabo la realización a nivel de diseño de los casos de uso relacionados con la gestión de la configuración del simulador.	
14.4.4		Realización de los casos de uso relacionados con el envío de tramas	Se llevará a cabo la realización a nivel de diseño de los casos de uso relacionados con la envío de tramas dentro del simulador.	
14.5	Implementación			
	14.5.1	Implementación de un prototipo interactivo	Se elaborará un prototipo interactivo que incluirá toda la interacción y componentes	

		<b>completo</b>	visuales del sistema.
	14.6	<b>Pruebas</b>	
	14.6.1	<b>Definición pruebas de integración</b>	Se definirán pruebas de integración, que serán utilizadas posteriormente para evaluar la fusión de distintos componentes del sistema.
15	<b>Hito fin iteración 2</b>		Este hito representa el fin de la segunda iteración de la fase de “Construcción”.
16	<b>Iteración 3</b>		Marca temporal que representa el comienzo de la iteración 3 de la fase de “Construcción”.
	16.1	<b>Modelado de negocio</b>	
	16.1.1	<b>Reunión con el cliente para revisar el desarrollo del proyecto</b>	Se realizará una reunión con el cliente para comentar el estado actual del desarrollo del proyecto.
	16.2	<b>Requisitos</b>	
	16.2.1	<b>Elaborar la matriz de rastreabilidad</b>	Se construirá la matriz de rastreabilidad, la cual reflejará las relaciones entre los distintos requisitos y objetivos identificados previamente
	16.3	<b>Análisis</b>	
	16.3.1	<b>Agrupación de las clases en los paquetes identificados</b>	Se clasificarán las distintas clases identificadas previamente en los paquetes previamente identificados.
	16.4	<b>Diseño</b>	
	16.4.1	<b>Realización de los casos de uso relacionados con la gestión de las capturas de tráfico</b>	Se llevará a cabo la realización a nivel de diseño de los casos de uso relacionados con el guardado y cargado de capturas de tráfico en el simulador.
	16.4.2	<b>Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i></b>	Se llevará a cabo la realización a nivel de diseño de los casos de uso relacionados con la aplicación de los distintos tipos de <i>timeouts</i> dentro del simulador.
	16.4.3	<b>Realización de los casos de uso relacionados con la recepción de tramas</b>	Se llevará a cabo la realización a nivel de diseño de los casos de uso relacionados con la recepción de tramas dentro del simulador.
	16.4.4	<b>Identificación y definición de interfaces</b>	Se identificarán y definirán las distintas interfaces utilizadas en el desarrollo del proyecto.
	16.5	<b>Implementación</b>	
	16.5.1	<b>Implementación de la interfaz gráfica</b>	Se implementarán todas las pantallas y ventanas de la aplicación, utilizando de guía el prototipo iterativo previamente realizado.
	16.5.2	<b>Implementación de la funcionalidad relacionada con la gestión de la configuración del simulador</b>	Se realizará la implementación de la funcionalidad relacionada con la gestión de la configuración del simulador.
	16.5.3	<b>Implementación de la funcionalidad relacionada con el envío de tramas</b>	Se realizará la implementación de la funcionalidad relacionada con el envío de tramas dentro del simulador.
	16.6	<b>Pruebas</b>	

		16.6.1	<b>Realización de pruebas de usabilidad</b>	Se realizarán pruebas de usabilidad con usuarios, utilizando el prototipo interactivo previamente desarrollado.
17	<b>Hito fin iteración 3</b>			Este hito representa el fin de la tercera iteración de la fase de "Construcción".
18	<b>Iteración 4</b>			Marca temporal que representa el comienzo de la iteración 4 de la fase de "Construcción".
	18.1	<b>Modelado de negocio</b>		
		18.1.1	<b>Investigación sobre posibles aplicaciones y utilidades del proyecto</b>	Se realizará una investigación sobre las posibles aplicaciones que tiene la aplicación desarrollada, así como sus utilidades.
	18.2	<b>Requisitos</b>		
		18.2.1	<b>Revisión de los requisitos</b>	Se realizará una revisión de los requisitos con el objetivo de encontrar y corregir posibles fallos que se hayan podido cometer en la especificación de requisitos.
	18.3	<b>Análisis</b>		
		18.3.1	<b>Elaboración de la propuesta de arquitectura</b>	Se elaborará una propuesta de arquitectura del sistema en el que se establecerán relaciones entre los distintos paquetes identificados previamente.
	18.4	<b>Diseño</b>		
		18.4.1	<b>Elaboración del diagrama de despliegue</b>	Se elaborará un diagrama de despliegue en el que se distribuirá la funcionalidad del sistema en nodos físicos.
	18.5	<b>Implementación</b>		
		18.5.1	<b>Implementación de la funcionalidad relacionada con la gestión de las capturas de tráfico</b>	Se realizará la implementación de la funcionalidad relacionada con el guardado y cargado de capturas de tráfico en el simulador.
		18.5.2	<b>Implementación de la funcionalidad relacionada con la aplicación de <i>timeouts</i></b>	Se realizará la implementación de la funcionalidad relacionada con la aplicación de los distintos tipos de <i>timeouts</i> dentro del simulador.
		18.5.3	<b>Implementación de la funcionalidad relacionada con la recepción de tramas</b>	Se realizará la implementación de la funcionalidad relacionada con la recepción de tramas dentro del simulador.
	18.6	<b>Pruebas</b>		
		18.6.1	<b>Realización de pruebas sobre la funcionalidad relacionada con la gestión de la configuración del simulador</b>	Se realizarán casos de prueba relacionados con la gestión de la configuración del simulador.
		18.6.2	<b>Realización de pruebas sobre la funcionalidad relacionada con el envío de tramas</b>	Se realizarán casos de prueba relacionados con el envío de tramas dentro del simulador.
19	<b>Hito fin iteración 4</b>			Este hito representa el fin de la cuarta iteración de la fase de "Construcción".

20	Hito fin fase “Construcción”		Este hito representa el fin de la fase de “Construcción”.
21	Transición		Marca temporal que representa el comienzo de la fase de “Transición”.
22	Iteración 1		Marca temporal que representa el comienzo de la iteración 1 de la fase de “Transición”.
	22.1	Modelado de negocio	
		22.1.1	Investigar posibles vías de mercado
	22.2	Requisitos	
		22.2.1	Contemplar la inclusión de nuevas funcionalidades
	22.3	Análisis	
		22.3.1	Análisis de posibles nuevas funcionalidades
	22.4	Diseño	
		22.4.1	Realización de la descripción de la arquitectura
	22.5	Implementación	
		22.5.1	Integración del sistema
	22.6	Pruebas	
22.6.1		Realización de pruebas sobre la funcionalidad relacionada con la gestión de las capturas de tráfico	Se realizarán casos de prueba relacionados con la gestión de las capturas de tráfico dentro del simulador.
22.6.2		Realización de pruebas sobre la funcionalidad relacionada con la aplicación de <i>timeouts</i>	Se realizarán casos de prueba relacionados con la aplicación de <i>timeouts</i> dentro del simulador.
22.6.3		Realización de pruebas sobre la funcionalidad relacionada con la recepción de tramas	Se realizarán casos de prueba relacionados con la recepción de tramas dentro del simulador
22.6.4		Realización de pruebas de	Se realizarán pruebas de integración para

			<b>integración</b>	comprobar que el sistema funciona correctamente en su conjunto.
23	<b>Hito fin iteración 1</b>			Este hito representa el fin de la primera iteración de la fase de “Transición”.
24	<b>Hito fin fase “Transición”</b>			Este hito representa el fin de la fase de “Transición”.
25	<b>Hito fin del proyecto</b>			Este hito representa el fin del desarrollo del proyecto.

Tabla 5: Descripción de tareas

## Identificación de hitos

Una vez se han identificado las distintas tareas en las que se descompone el desarrollo del proyecto, y se han distribuido estas tareas en las distintas disciplinas e iteraciones, el siguiente paso es definir correctamente el paso de una iteración a otra.

En el Proceso Unificado, para marcar la finalización de una iteración se hace uso de un hito. Este hito marca la finalización de un conjunto de tareas asociadas a una iteración y el comienzo de otro conjunto de tareas de la siguiente iteración. Estos hitos son muy importantes ya que nos ayudaran a separar las iteraciones y nos permiten controlar la dirección y progreso del desarrollo.

Es muy importante tener en cuenta que los hitos, a diferencia de las tareas, no consumen ni tiempo ni recursos ya que son una marca temporal que separa las tareas de varias iteraciones. Debido a esto, estos hitos en nuestra planificación temporal aparecerán con duración 0 y no se le asigna ningún recurso.

Un ejemplo de hito definido en la herramienta *Windows Project* puede ser el siguiente:

Fin iteración 1	0 días	mié 07/11/18	mié 07/11/18 3
-----------------	--------	--------------	----------------

Figura 8: Ejemplo de hito en *Windows Project*

Anteriormente en la identificación y descripción de tareas, se incluyeron los hitos secundarios los cuales marcan el final de una iteración como los hitos principales, que marcan el final de una fase. De esta manera, la identificación de los hitos se encuentra correctamente realizada.



## Estimación de la duración de las tareas

Una vez se han identificado las tareas a realizar, se han agrupado estas tareas dentro de las distintas iteraciones del desarrollo del proyecto y se ha realizado la asignación de recursos a estas tareas, ahora se realizará la estimación de la duración temporal de cada una de estas tareas.

Para la estimación de las tareas se tendrá en cuenta el valor de la estimación de esfuerzo obtenido anteriormente, de manera que la estimación de la duración de las tareas vaya en consecuencia a la estimación realizada anteriormente.

En la siguiente tabla, se mostrará la estimación de la duración de cada una de las tareas identificada anteriormente, así como la asignación de recursos a cada una de las tareas (en este caso, al tener solo un miembro en el equipo de desarrollo la asignación de tareas siempre será del 100%):

ID	Nombre	Asignación temporal	Asignación de recursos
2.1.1	Investigar sobre el funcionamiento teórico del protocolo HDLC	3 días	100%
2.1.2	Investigar proyectos similares existentes	6 días	100%
2.2.1	Reuniones con el cliente	0.5 días	100%
2.2.2	Identificación y definición de los primeros objetivos	0.5 días	100%
2.2.3	Determinar el ámbito del sistema	0.5 días	100%
2.3.1	Realización de un primer boceto de la arquitectura	1 día	100%
2.4.1	Distribución de los subsistemas	0.5 días	100%
2.5.1	Descarga e instalación del entorno de desarrollo	0.1 días	100%
2.6.1	Pruebas sobre el correcto funcionamiento del entorno de desarrollo	0.2 días	100%
6.1.1	Investigación sobre técnicas de comunicación entre procesos	3 días	100%
6.2.1	Identificación de actores	0.25 días	100%
6.2.2	Identificar requisitos no funcionales	0.5 días	100%
6.2.3	Identificar requisitos de información	2 días	100%
6.2.4	Identificación de los paquetes funcionales	1 día	100%
6.3.1	Identificación de las primeras clases entidad	1 día	100%
6.4.1	Definición general del mapa del sitio	0.5 días	100%
6.5.1	Elaboración de un primer prototipo	2 días	100%
6.6.1	Realizar pruebas en entornos similares	2 días	100%
8.1.1	Investigación sobre la creación de hilos en el lenguaje de programación utilizado	3 días	100%
8.2.1	Identificación de casos de uso relacionados con la gestión de la configuración del simulador	1 día	100%

8.2.2	Identificación de casos de uso relacionados con el envío de tramas	3 días	100%
8.2.3	Realización de encuestas de contenido a los usuarios	2 días	100%
8.3.1	Identificación del resto de clases entidad y las relaciones entre ellas	1 día	100%
8.3.2	Identificación de los primeros paquetes	0.5 días	100%
8.4.1	Evaluación de decisiones de diseño de la interfaz gráfica	1 día	100%
8.5.1	Implementación de un modelo de comunicación entre 2 procesos	1 día	100%
8.6.1	Definición de casos de prueba	2 días	100%
12.1.1	Identificación de riesgos	0.5 días	100%
12.2.1	Identificación de casos de uso relacionados con la gestión de las capturas de tráfico	1 día	100%
12.2.2	Identificación de casos de uso relacionados con la aplicación de <i>timeouts</i>	1 día	100%
12.2.3	Identificación de casos de uso relacionados con la recepción de tramas	2 días	100%
12.3.1	Identificación de clases de control y clases interfaz	2 días	100%
12.3.2	Identificación del resto de paquetes	1 día	100%
12.3.3	Realización de los casos de uso relacionados con la gestión de la configuración del simulador	1 día	100%
12.3.4	Realización de los casos de uso relacionados con el envío de tramas	3 días	100%
12.4.1	Identificación de los primeros subsistemas	0.5 días	100%
12.4.2	Elaboración de un esquema de la visualización de la interfaz gráfica	2 días	100%
12.4.3	Identificación las primeras clases de diseño	1 día	100%
12.5.1	Implementación de hilos que realicen tareas simples y recursivas	1 día	100%
12.6.1	Definir una estrategia para la realización de pruebas	0.5 días	100%
14.1.1	Priorización de riesgos	0.25 días	100%
14.2.1	Identificación de escenarios de uso	3 días	100%
14.2.2	Elaboración del diagrama de casos de uso	1 día	100%
14.3.1	Realización de los casos de uso relacionados con la gestión de las capturas de tráfico	1 día	100%
14.3.2	Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i>	1 día	100%
14.3.3	Realización de los casos de uso relacionados con la recepción de tramas	2 días	100%
14.3.4	Análisis de tareas	3 días	100%
14.4.1	Identificación del resto de clases de diseño	2 días	100%
14.4.2	Identificación de resto de subsistemas de diseño	0.5 días	100%

14.4.3	Realización de los casos de uso relacionados con la gestión de la configuración del simulador	1 día	100%
14.4.4	Realización de los casos de uso relacionados con el envío de tramas	4 días	100%
14.5.1	Implementación de un prototipo interactivo completo	6 días	100%
14.6.1	Definición pruebas de integración	0.5 días	100%
16.1.1	Reunión con el cliente para revisar el desarrollo del proyecto	0.25 días	100%
16.2.1	Elaborar la matriz de rastreabilidad	0.25 días	100%
16.3.1	Agrupación de las clases en los paquetes identificados	0.25 días	100%
16.4.1	Realización de los casos de uso relacionados con la gestión de las capturas de tráfico	1 día	100%
16.4.2	Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i>	1 día	100%
16.4.3	Realización de los casos de uso relacionados con la recepción de tramas	3 días	100%
16.4.4	Identificación y definición de interfaces	1 día	100%
16.5.1	Implementación de la interfaz gráfica	5 días	100%
16.5.2	Implementación de la funcionalidad relacionada con la gestión de la configuración del simulador	3 días	100%
16.5.3	Implementación de la funcionalidad relacionada con el envío de tramas	8 días	100%
16.6.1	Realización de pruebas de usabilidad	2 días	100%
18.1.1	Investigación sobre posibles aplicaciones y utilidades del proyecto	0.25 días	100%
18.2.1	Revisión de los requisitos	0.5 días	100%
18.3.1	Elaboración de la propuesta de arquitectura	0.25 días	100%
18.4.1	Elaboración del diagrama de despliegue	0.25 días	100%
18.5.1	Implementación de la funcionalidad relacionada con la gestión de las capturas de tráfico	3 días	100%
18.5.2	Implementación de la funcionalidad relacionada con la aplicación de <i>timeouts</i>	5 días	100%
18.5.3	Implementación de la funcionalidad relacionada con la recepción de tramas	6 días	100%
18.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de la configuración del simulador	0.25 días	100%
18.6.2	Realización de pruebas sobre la funcionalidad relacionada con el envío de tramas	0.5 días	100%
22.1.1	Investigar posibles vías de mercado	0.25 días	100%
22.2.1	Contemplar la inclusión de nuevas funcionalidades	0.5 días	100%
22.3.1	Análisis de posibles nuevas funcionalidades	1 día	100%
22.4.1	Realización de la descripción de la arquitectura	0.25 días	100%

22.5.1	Integración del sistema	7 días	100%
22.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de las capturas de tráfico	0.5 días	100%
22.6.2	Realización de pruebas sobre la funcionalidad relacionada con la aplicación de <i>timeouts</i>	0.5 días	100%
22.6.3	Realización de pruebas sobre la funcionalidad relacionada con la recepción de tramas	0.5 días	100%
22.6.4	Realización de pruebas de integración	0.5 días	100%

Tabla 6: Estimación de las tareas

A modo de resumen, en la siguiente tabla se mostrará la duración estimada de cada una de las iteraciones junto con la estimación temporal del proyecto completo.

ID	Nombre	Asignación temporal
0	Inicio del proyecto	<b><u>135.05 días</u></b>
1	Inicio	<b>12.3 días</b>
2	Iteración 1	12.3 días
5	Elaboración	<b>29.25 días</b>
6	Iteración 1	11.25 días
8	Iteración 2	17 días
11	Construcción	<b>82.5 días</b>
12	Iteración 1	16.5 días
14	Iteración 2	25.25 días
16	Iteración 3	24.75 días
18	Iteración 4	16 días
21	Transición	<b>11 días</b>
22	Iteración 1	11 días

Tabla 7: Tabla resumen estimación iteraciones del proyecto

## Asignación de recursos

Una vez se han identificado las tareas a realizar y se han agrupado estas tareas dentro de las distintas iteraciones del desarrollo del proyecto, ahora se asignarán recursos humanos (trabajadores) a cada una de estas tareas.

Para realizar esta asignación de recursos a las tareas hay que tener en cuenta la disponibilidad de los trabajadores y de no sobrecargar a los trabajadores con más tareas de las que puedan realizar.

En el caso particular del sistema a desarrollar, la asignación de recursos a las tareas será bastante trivial ya que el equipo de desarrollo está formado por un único miembro, por lo que la responsabilidad de todas las tareas recaerá en su totalidad al único desarrollador.

## Dependencias entre tareas

Una vez se han identificado las tareas a realizar y se han asignado tiempo y recursos a estas tareas, el siguiente y último paso es definir las dependencias o relaciones entre las tareas. Estas relaciones van a impedir que podamos realizar todas las tareas de una interacción de forma paralela ya que es necesario una tarea o conjunto de tareas se haya realizado para poder realizar otra tareas o conjunto de tareas.

Existen 4 tipos de dependencias entre tareas que se pueden especificar en Windows:

Partimos del hecho de que se tiene una dependencia entre dos tareas A y B

- **Fin a comienzo (FC):** La tarea dependiente B no puede comenzar hasta que se complete la tarea de la que depende A.
- **Comienzo a comienzo (CC):** La tarea dependiente B no puede comenzar hasta que comience la tarea de la que depende A.
- **Fin a fin (FF):** La tarea dependiente B no puede completarse hasta que se complete la tarea de la que depende A.
- **Comienzo a fin (CF):** La tarea dependiente B no puede completarse hasta que comience la tarea de la que depende A.

En el caso particular de este proyecto, se realizará una planificación lineal en la que solo una tarea puede ser realizada si la tarea anterior ha sido finalizada con éxito. De esta manera, las dependencias utilizadas para implementar este tipo de planificación serán las dependencias fin a comienzo (FC), donde todas las tareas a excepción de la primera tendrán una dependencia fin a comienzo (FC) con la tarea anterior.

En la siguiente tabla se mostrarán las distintas dependencias existentes entre las tareas indicado para cada tarea, el nombre e identificador de la tarea de la que depende:

Tarea		Dependencia	
2.1.1	Investigar sobre el funcionamiento teórico del protocolo HDLC	-	-
2.1.2	Investigar proyectos similares existentes	2.1.1	Investigar sobre el funcionamiento teórico del protocolo HDLC
2.2.1	Reuniones con el cliente	2.1.2	Investigar proyectos similares existentes
2.2.2	Identificación y definición de los primeros objetivos	2.2.1	Reuniones con el cliente
2.2.3	Determinar el ámbito del sistema	2.2.2	Identificación y definición de los primeros objetivos
2.3.1	Realización de un primer boceto de la arquitectura	2.2.3	Determinar el ámbito del sistema
2.4.1	Distribución de los subsistemas	2.3.1	Realización de un primer boceto de la arquitectura
2.5.1	Descarga e instalación del entorno de desarrollo	2.4.1	Distribución de los subsistemas

2.6.1	Pruebas sobre el correcto funcionamiento del entorno de desarrollo	2.5.1	Descarga e instalación del entorno de desarrollo
6.1.1	Investigación sobre técnicas de comunicación entre procesos	2.6.1	Pruebas sobre el correcto funcionamiento del entorno de desarrollo
6.2.1	Identificación de actores	6.1.1	Investigación sobre técnicas de comunicación entre procesos
6.2.2	Identificar requisitos no funcionales	6.2.1	Identificación de actores
6.2.3	Identificar requisitos de información	6.2.2	Identificar requisitos no funcionales
6.2.4	Identificación de los paquetes funcionales	6.2.3	Identificar requisitos de información
6.3.1	Identificación de las primeras clases entidad	6.2.4	Identificación de los paquetes funcionales
6.4.1	Definición general del mapa del sitio	6.3.1	Identificación de las primeras clases entidad
6.5.1	Elaboración de un primer prototipo	6.4.1	Definición general del mapa del sitio
6.6.1	Realizar pruebas en entornos similares	6.5.1	Elaboración de un primer prototipo
8.1.1	Investigación sobre la creación de hilos en el lenguaje de programación utilizado	6.6.1	Realizar pruebas en entornos similares
8.2.1	Identificación de casos de uso relacionados con la gestión de la configuración del simulador	8.1.1	Investigación sobre la creación de hilos en el lenguaje de programación utilizado
8.2.2	Identificación de casos de uso relacionados con el envío de tramas	8.2.1	Identificación de casos de uso relacionados con la gestión de la configuración del simulador
8.2.3	Realización de encuestas de contenido a los usuarios	8.2.2	Identificación de casos de uso relacionados con el envío de tramas
8.3.1	Identificación del resto de clases entidad y las relaciones entre ellas	8.2.3	Realización de encuestas de contenido a los usuarios
8.3.2	Identificación de los primeros paquetes	8.3.1	Identificación del resto de clases entidad y las relaciones entre ellas
8.4.1	Evaluación de decisiones de diseño de la interfaz gráfica	8.3.2	Identificación de los primeros paquetes
8.5.1	Implementación de un modelo de comunicación entre 2 procesos	8.4.1	Evaluación de decisiones de diseño de la interfaz gráfica
8.6.1	Definición de casos de prueba	8.5.1	Implementación de un modelo de comunicación entre 2 procesos
12.1.1	Identificación de riesgos	8.6.1	Definición de casos de prueba
12.2.1	Identificación de casos de uso relacionados con la gestión de las capturas de tráfico	12.1.1	Identificación de riesgos
12.2.2	Identificación de casos de uso relacionados con la aplicación de	12.2.1	Identificación de casos de uso relacionados con la gestión de

	<b><i>timeouts</i></b>		<b>las capturas de tráfico</b>
12.2.3	<b>Identificación de casos de uso relacionados con la recepción de tramas</b>	12.2.2	<b>Identificación de casos de uso relacionados con la aplicación de <i>timeouts</i></b>
12.3.1	<b>Identificación de clases de control y clases interfaz</b>	12.2.3	<b>Identificación de casos de uso relacionados con la recepción de tramas</b>
12.3.2	<b>Identificación del resto de paquetes</b>	12.3.1	<b>Identificación de clases de control y clases interfaz</b>
12.3.3	<b>Realización de los casos de uso relacionados con la gestión de la configuración del simulador</b>	12.3.2	<b>Identificación del resto de paquetes</b>
12.3.4	<b>Realización de los casos de uso relacionados con el envío de tramas</b>	12.3.3	<b>Realización de los casos de uso relacionados con la gestión de la configuración del simulador</b>
12.4.1	<b>Identificación de los primeros subsistemas</b>	12.3.4	<b>Realización de los casos de uso relacionados con el envío de tramas</b>
12.4.2	<b>Elaboración de un esquema de la visualización de la interfaz gráfica</b>	12.4.1	<b>Identificación de los primeros subsistemas</b>
12.4.3	<b>Identificación las primeras clases de diseño</b>	12.4.2	<b>Elaboración de un esquema de la visualización de la interfaz gráfica</b>
12.5.1	<b>Implementación de hilos que realicen tareas simples y recursivas</b>	12.4.3	<b>Identificación las primeras clases de diseño</b>
12.6.1	<b>Definir una estrategia para la realización de pruebas</b>	12.5.1	<b>Implementación de hilos que realicen tareas simples y recursivas</b>
14.1.1	<b>Priorización de riesgos</b>	12.6.1	<b>Definir una estrategia para la realización de pruebas</b>
14.2.1	<b>Identificación de escenarios de uso</b>	14.1.1	<b>Priorización de riesgos</b>
14.2.2	<b>Elaboración del diagrama de casos de uso</b>	14.2.1	<b>Identificación de escenarios de uso</b>
14.3.1	<b>Realización de los casos de uso relacionados con la gestión de las capturas de tráfico</b>	14.2.2	<b>Elaboración del diagrama de casos de uso</b>
14.3.2	<b>Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i></b>	14.3.1	<b>Realización de los casos de uso relacionados con la gestión de las capturas de tráfico</b>
14.3.3	<b>Realización de los casos de uso relacionados con la recepción de tramas</b>	14.3.2	<b>Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i></b>
14.3.4	<b>Análisis de tareas</b>	14.3.3	<b>Realización de los casos de uso relacionados con la recepción de tramas</b>
14.4.1	<b>Identificación del resto de clases de diseño</b>	14.3.4	<b>Análisis de tareas</b>
14.4.2	<b>Identificación de resto de subsistemas de diseño</b>	14.4.1	<b>Identificación del resto de clases de diseño</b>
14.4.3	<b>Realización de los casos de uso relacionados con la gestión de la configuración del simulador</b>	14.4.2	<b>Identificación de resto de subsistemas de diseño</b>



14.4.4	Realización de los casos de uso relacionados con el envío de tramas	14.4.3	Realización de los casos de uso relacionados con la gestión de la configuración del simulador
14.5.1	Implementación de un prototipo interactivo completo	14.4.4	Realización de los casos de uso relacionados con el envío de tramas
14.6.1	Definición pruebas de integración	14.5.1	Implementación de un prototipo interactivo completo
16.1.1	Reunión con el cliente para revisar el desarrollo del proyecto	14.6.1	Definición pruebas de integración
16.2.1	Elaborar la matriz de rastreabilidad	16.1.1	Reunión con el cliente para revisar el desarrollo del proyecto
16.3.1	Agrupación de las clases en los paquetes identificados	16.2.1	Elaborar la matriz de rastreabilidad
16.4.1	Realización de los casos de uso relacionados con la gestión de las capturas de tráfico	16.3.1	Agrupación de las clases en los paquetes identificados
16.4.2	Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i>	16.4.1	Realización de los casos de uso relacionados con la gestión de las capturas de tráfico
16.4.3	Realización de los casos de uso relacionados con la recepción de tramas	16.4.2	Realización de los casos de uso relacionados con la aplicación de <i>timeouts</i>
16.4.4	Identificación y definición de interfaces	16.4.3	Realización de los casos de uso relacionados con la recepción de tramas
16.5.1	Implementación de la interfaz gráfica	16.4.4	Identificación y definición de interfaces
16.5.2	Implementación de la funcionalidad relacionada con la gestión de la configuración del simulador	16.5.1	Implementación de la interfaz gráfica
16.5.3	Implementación de la funcionalidad relacionada con el envío de tramas	16.5.2	Implementación de la funcionalidad relacionada con la gestión de la configuración del simulador
16.6.1	Realización de pruebas de usabilidad	16.5.3	Implementación de la funcionalidad relacionada con el envío de tramas
18.1.1	Investigación sobre posibles aplicaciones y utilidades del proyecto	16.6.1	Realización de pruebas de usabilidad
18.2.1	Revisión de los requisitos	18.1.1	Investigación sobre posibles aplicaciones y utilidades del proyecto
18.3.1	Elaboración de la propuesta de arquitectura	18.2.1	Revisión de los requisitos
18.4.1	Elaboración del diagrama de despliegue	18.3.1	Elaboración de la propuesta de arquitectura
18.5.1	Implementación de la funcionalidad relacionada con la gestión de las capturas de tráfico	18.4.1	Elaboración del diagrama de despliegue

18.5.2	Implementación de la funcionalidad relacionada con la aplicación de <i>timeouts</i>	18.5.1	Implementación de la funcionalidad relacionada con la gestión de las capturas de tráfico
18.5.3	Implementación de la funcionalidad relacionada con la recepción de tramas	18.5.2	Implementación de la funcionalidad relacionada con la aplicación de <i>timeouts</i>
18.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de la configuración del simulador	18.5.3	Implementación de la funcionalidad relacionada con la recepción de tramas
18.6.2	Realización de pruebas sobre la funcionalidad relacionada con el envío de tramas	18.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de la configuración del simulador
22.1.1	Investigar posibles vías de mercado	18.6.2	Realización de pruebas sobre la funcionalidad relacionada con el envío de tramas
22.2.1	Contemplar la inclusión de nuevas funcionalidades	22.1.1	Investigar posibles vías de mercado
22.3.1	Análisis de posibles nuevas funcionalidades	22.2.1	Contemplar la inclusión de nuevas funcionalidades
22.4.1	Realización de la descripción de la arquitectura	22.3.1	Análisis de posibles nuevas funcionalidades
22.5.1	Integración del sistema	22.4.1	Realización de la descripción de la arquitectura
22.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de las capturas de tráfico	22.5.1	Integración del sistema
22.6.2	Realización de pruebas sobre la funcionalidad relacionada con la aplicación de <i>timeouts</i>	22.6.1	Realización de pruebas sobre la funcionalidad relacionada con la gestión de las capturas de tráfico
22.6.3	Realización de pruebas sobre la funcionalidad relacionada con la recepción de tramas	22.6.2	Realización de pruebas sobre la funcionalidad relacionada con la aplicación de <i>timeouts</i>
22.6.4	Realización de pruebas de integración	22.6.3	Realización de pruebas sobre la funcionalidad relacionada con la recepción de tramas

Tabla 8: Tabla de dependencias entre tareas

## Diagrama de Gantt

Para concluir la planificación temporal del proyecto, se elaborará el diagrama de Gantt de la planificación realizada. El diagrama de Gantt permite representar gráficamente el tiempo empleado para realizar las tareas dentro de un calendario de trabajo, así como sus dependencias y recursos asignados a dichas tareas.

Para elaborar el diagrama de Gantt, se utilizará la herramienta *Windows Project*, la cual realiza este cometido de manera automática. Sin embargo, debe crearse anteriormente un proyecto en el que se introduzcan las tareas identificadas agrupadas en iteraciones, la asignación de recursos a dichas tareas y las dependencias entre las tareas.

Una vez se proporciona a *Windows Project* toda la información necesaria sobre el proyecto a desarrollar, se obtendrá representado el diagrama de Gantt, el cual se mostrará en las siguientes figuras:

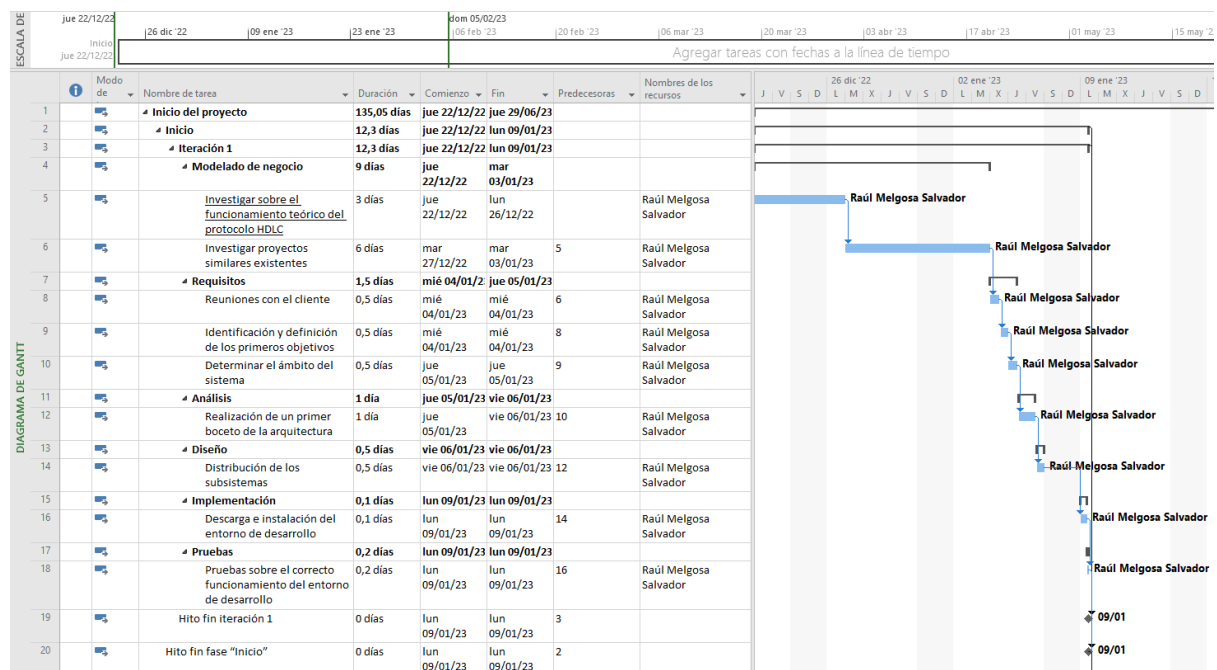


Figura 9: Diagrama de Gantt y Planificación Temporal (1)



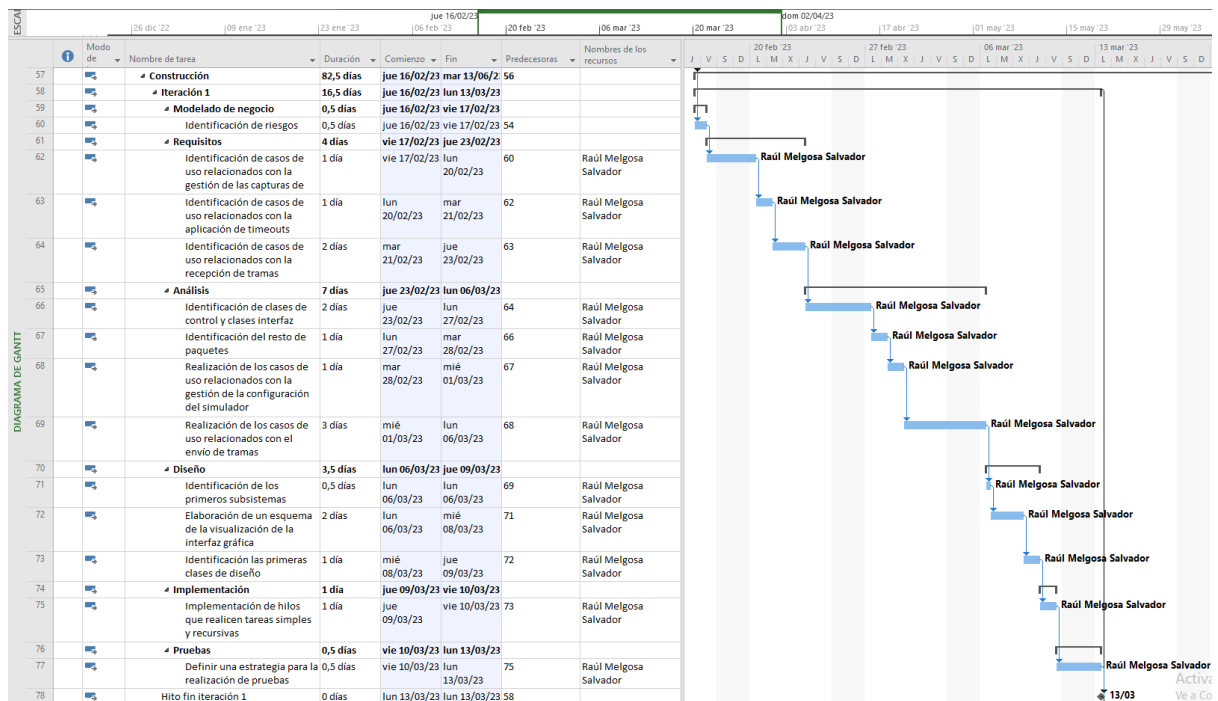


Figura 12: Diagrama de Gantt y Planificación Temporal (4)

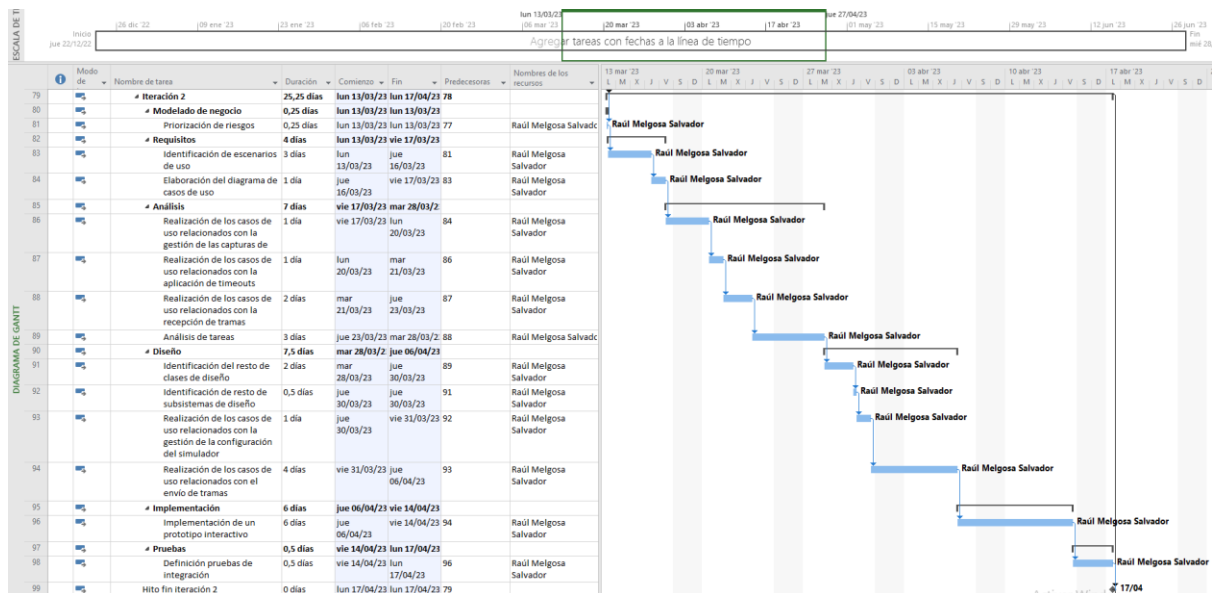


Figura 13: Diagrama de Gantt y Planificación Temporal (5)

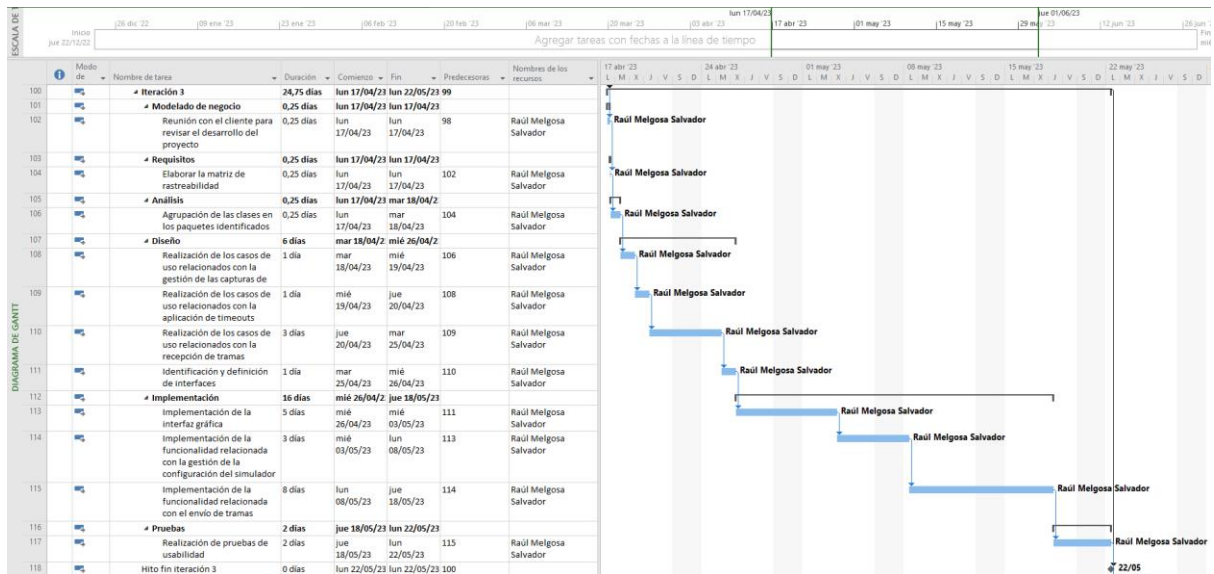


Figura 14: Diagrama de Gantt y Planificación Temporal (6)

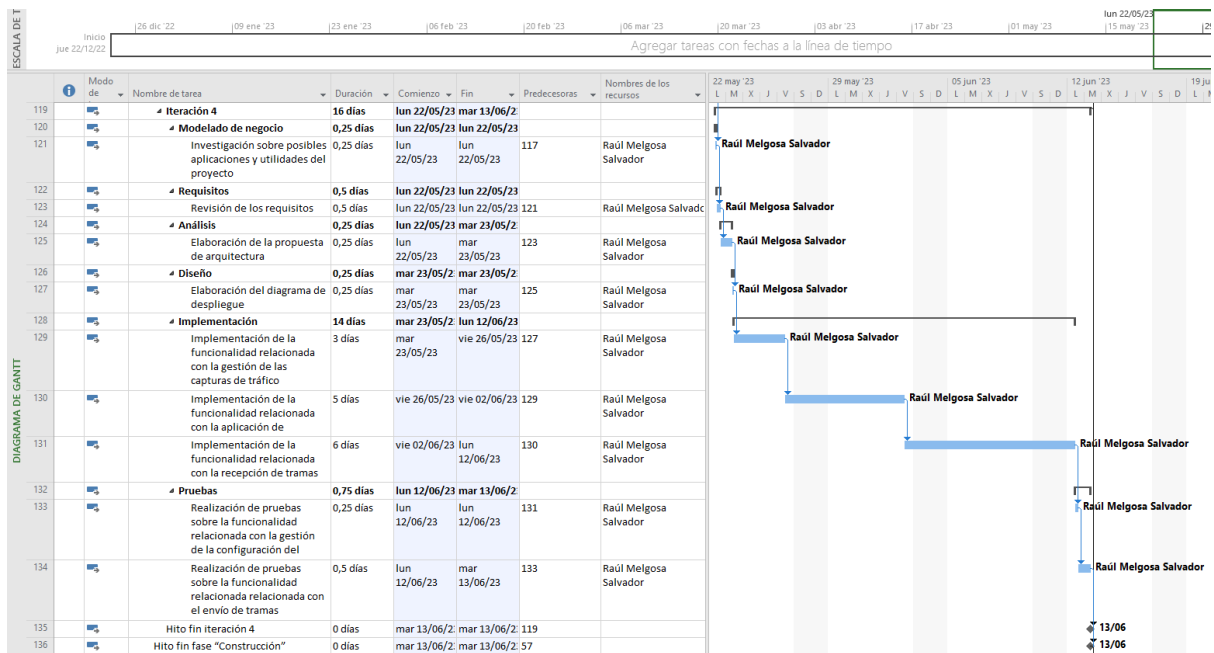


Figura 15: Diagrama de Gantt y Planificación Temporal (7)

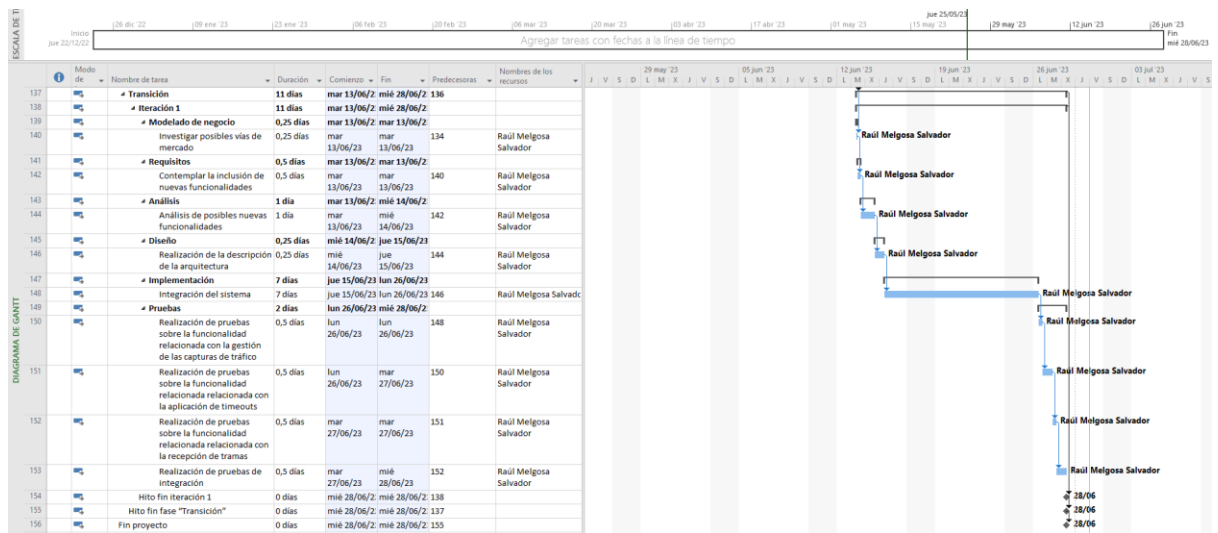


Figura 16: Diagrama de Gantt y Planificación Temporal (8)

# Conclusiones

Tras realizar la estimación del esfuerzo como la planificación temporal, se pueden obtener las siguientes conclusiones:

- La estimación del esfuerzo necesario para la realización del proyecto en su totalidad es de 1107 horas, las cuales tendrán que ser repartidas entre el único miembro del equipo de desarrollo.
- En la planificación temporal, se ha obtenido una estimación total de 135 días de trabajo. Teniendo en cuenta que las jornadas de trabajo son de 8 horas, la estimación total de la duración de las tareas será de 1080 horas. Esta cifra es muy cercana a la cifra obtenida en la estimación del esfuerzo, lo cual es totalmente coherente y lógico.
- Siguiendo la planificación temporal, la fecha en la que se estima la finalización del proyecto es el día 29 de junio de 2023. La fecha de entrega del proyecto es el 5 de julio de 2023 por lo que se garantizan una serie de días para corregir fallos y para cubrir retrasos con la planificación original.



# Bibliografía

- "Practica 1: Estimación del esfuerzo", María N. Moreno García,  
[https://studium22.usal.es/pluginfile.php/55969/mod\\_resource/content/1/GP-GII\\_Primer\\_practica.pdf](https://studium22.usal.es/pluginfile.php/55969/mod_resource/content/1/GP-GII_Primer_practica.pdf).
- "Practica 2: Planificación temporal", María N. Moreno García, María Navarro Cáceres,  
[https://studium22.usal.es/pluginfile.php/55987/mod\\_resource/content/2/Pra%CC%81ctica%202-Planificacio%CC%81n%20temporal.pdf](https://studium22.usal.es/pluginfile.php/55987/mod_resource/content/2/Pra%CC%81ctica%202-Planificacio%CC%81n%20temporal.pdf).
- "EZ Estimate".
- "Microsoft Project" Microsoft, <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>.