

# SALTOUR

Soluciones tecnológicas para una mejora de la  
experiencia turística

## Anexo I: Planificación temporal



VNiVERSIDAD  
D SALAMANCA

Proyecto de Fin de Máster en Ingeniería Informática

**Tutores:**

Ángel Luis Sánchez Lázaro

Ana Belén Gil González

**Alumno:**

Miguel Cabezas Puerto



## Contenido

<b>1. Introducción .....</b>	<b>5</b>
<b>2. Medición del esfuerzo .....</b>	<b>5</b>
2.1 Identificar actores y asignarles su complejidad.....	6
2.2 Identificar casos de uso y asignarles su complejidad.....	6
3.2 Determinar la complejidad de los factores técnicos y del entorno .....	24
3.3 Conclusión .....	26
<b>3. Planificación temporal .....</b>	<b>27</b>
3.1 Calendario de trabajo.....	28
3.2 Esquema de planificación temporal .....	28
3.3 Uso de recursos.....	35

Ilustración 1 Estimación esfuerzo EZEstimate .....	27
Ilustración 2 Fases Proceso Unificado .....	28
Ilustración 3 Tareas .....	34
Ilustración 4 Uso de recursos .....	35

## 1. Introducción

Este primer anexo contiene la medición del esfuerzo, así como la planificación temporal que se ha seguido en el desarrollo del proyecto.

## 2. Medición del esfuerzo

Una vez definidos los casos de uso y actores del sistema se ha estimado el esfuerzo requerido en el desarrollo de este a partir de la métrica de los puntos de casos de uso mediante la herramienta EZEstimate.

Esta métrica se utiliza para evaluar la funcionalidad representada en forma de casos de uso y tiene en cuenta actores, escenarios, factores técnicos y del entorno. Requiere del cálculo de tres variables:

- ◇ UUCP (Unadjusted Use Case Points)
  - UUCW (Unadjusted Use Case Weight): Considera el número y complejidad de los casos de uso (número de pasos de los escenarios, complejidad de la interfaz de usuario, número de entidades de la base de datos a las que accede y número de clases implicadas)
  - UAW (Unadjusted Actor Weight): Considera el número y complejidad de los actores
    - 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
    - Complejo: Persona con una interfaz gráfica
- ◇ TCF: factores de complejidad técnica, son trece, se verán más adelante.
- ◇ ECF: factores de complejidad del entorno, son ocho, se verán más adelante.

Así, los puntos de casos de uso se calculan como  $UCP = UUCP * TCF * ECF$  y el esfuerzo como:

$$Esfuerzo = UCP * F$$

Siendo F el factor de conversión (número de horas de persona por UCP)

Todos estos cálculos los realiza la herramienta, para ello se ha seguido el siguiente proceso.

- 1) Crear un módulo por cada paquete del modelo de casos de uso del sistema
- 2) Identificar actores y asignarles su complejidad
- 3) Identificar casos de uso y asignarles su complejidad en función del número de transacciones
  - ◇ Simple: 3 transacciones o menos
  - ◇ Medio: De 4 a 7 transacciones
  - ◇ Complejo: Más de 7 transacciones
- 4) Determinar la influencia (complejidad percibida) de los factores técnicos y de los factores de entorno asignándole un valor entre 0 y 5
- 5) Asignar el número de horas de persona por punto de caso de uso.

### 2.1 Identificar actores y asignarles su complejidad

En base a los actores identificados en la elicitación de requisitos en el Anexo II, se procede a asignarles una complejidad.

- ◇ Usuario: compleja debido a que la interacción de este con la aplicación debe ser a través de interfaces gráficas.
- ◇ Sistema: medio debido a que se comunica con Internet para acceder a la plataforma de Google, así como para otras funcionalidades como mandar correos, compartir la aplicación, etc.

### 2.2 Identificar casos de uso y asignarles su complejidad

En base a los casos de uso identificados en la elicitación de requisitos recogidos en el Anexo II, se procede a asignarles una complejidad en función del número de transacciones. Se entiende como transacción cuando el actor, en este caso Usuario, realiza alguna acción y el sistema realiza algo respecto al caso de uso. En caso de incluir otro caso de uso no se contabiliza este como transacción.

<b>UC-0001</b>	<b>Guardar tiempo reto</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	

<b>Dependencias</b>	OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos IRQ-0001 Usuario IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0004 Tiempo de respuesta UC-0004 Detener reto	
<b>Descripción</b>	El sistema debe ser capaz de almacenar el tiempo que lleva el usuario intentando conseguir un reto	
<b>Precondición</b>	El sistema debe haber detenido el tiempo	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-02 ( <i>Sistema</i> ) transforma los datos de tiempo del usuario para el reto activo y los almacena de forma persistente.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Si no existe conexión a Internet el sistema emite un mensaje y no almacena nada.
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que no cuenta con ninguna transacción al realizar solo el sistema acciones.

<b>UC-0002</b>	<b>Iniciar reto</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	

<b>Dependencias</b>	OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos OBJ-0004 Localizar puntos de interés IRQ-0001 Usuario IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas	
<b>Descripción</b>	El sistema debe ser capaz de empezar a contar el tiempo de compleción de un reto	
<b>Precondición</b>	Debe existir dicha oferta	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>normal</b>	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de Iniciar un reto.
	2	El sistema marca el reto como activo y comienza a contar el tiempo
	3	El sistema devuelve la información de localización del reto IRQ-0002 (Reto)
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que cuenta con una única transacción en la que el usuario selecciona una opción y el sistema realiza dos acciones.

<b>UC-0003</b>	<b>Mostrar reto en mapa</b>
<b>Versión</b>	1.0 (01/03/2022)

<b>Autores</b>	Miguel Cabezas Puerto	
<b>Fuentes</b>		
<b>Dependencias</b>	OBJ-0003 Gestionar retos OBJ-0004 Localizar puntos de interés IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0002      Iniciar reto	
<b>Descripción</b>	El sistema debe permitir la localización de los puntos de interés representados por los retos	
<b>Precondición</b>	Debe existir dicha oferta	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 (Usuario) realiza el UC-0002 (Iniciar reto)
	2	El sistema ubica el reto utilizando sus coordenadas

	3	El sistema devuelve la información de localización del reto correspondiente.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si la ubicación de la oferta no es válida o se produce algún error se le notifica al usuario y termina el caso de uso.
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que solo contiene una transacción, cuando el usuario trata de iniciar un reto y el sistema lo ubica sobre el mapa y comienza el conteo de tiempo

<b>UC-0004</b>	<b>Detener reto</b>	
<b>Versión</b>	1.0 (01/03/2022)	
<b>Autores</b>	Miguel Cabezas Puerto	
<b>Fuentes</b>		
<b>Dependencias</b>	OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos IRQ-0001 Usuario IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0002 Iniciar reto	
<b>Descripción</b>	El sistema debe ser capaz de terminar de contar el tiempo de compleción de un reto	
<b>Precondición</b>	Debe estar el reto activo e iniciado	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de

		detener un reto
	2	El sistema marca el reto como no activo y detiene el tiempo
<b>Postcondición</b>	UC-0001 Guardar información reto	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que solo contiene una transacción en la que el usuariodetiene el reto y el sistema detiene el conteo.

<b>UC-0005</b>	<b>Escanear QR</b>	
<b>Versión</b>	1.0 (01/03/2022)	
<b>Autores</b>	Miguel Cabezas Puerto	
<b>Fuentes</b>		
<b>Dependencias</b>	OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos IRQ-0001 Usuario IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0002 Iniciar reto	
<b>Descripción</b>	El sistema debe ser capaz de terminar de contar el tiempo de compleción de un reto	
<b>Precondición</b>	Debe estar el reto activo	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de Escanear QR
	2	El sistema comprueba que se trata

		del QR asociado al reto activo, marca el reto como no activo completado y detiene el tiempo
--	--	---

<b>Postcondición</b>	UC-0001 Guardar información reto	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	En caso de no tratarse del QR asociado al reto activo el sistema emite un mensaje al usuario y continúa con la cuenta del tiempo dejando el reto en el mismo estado de procedencia (activo)
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene una sola transacción en la que el usuario indica que quiere escanear un QR y el sistema comprueba el código escaneado.

<b>UC-0006</b>	<b>Registrar usuario</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	
<b>Dependencias</b>	OBJ-0002 Gestionar información de usuario IRQ-0001 Usuario NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma
<b>Descripción</b>	El sistema debe permitir el registro de un nuevo usuario

<b>Precondición</b>	Ninguna	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de registrar nuevo usuario
	2	El sistema solicita la información al usuario, en particular la indicada en el IRQ-0001 ( <i>Usuario</i> ).
	3	El usuario introduce los datos correspondientes
	4	El sistema almacena los datos.
	5	El usuario confirma el registro.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el sistema detecta algún error en la validez de los datos volverá al paso 2.
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene dos transacciones de intercambio de datos entre usuario y sistema.

<b>UC-0007</b>	<b>Login usuario</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	
<b>Dependencias</b>	OBJ-0002 Gestionar información de usuario IRQ-0001 Usuario NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0006 Registrar usuario

<b>Descripción</b>	El sistema debe permitir la autenticación de un usuario
--------------------	---

<b>Precondición</b>	Ninguna	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de ingresar en la aplicación
	2	El sistema solicita la información al usuario, en particular la indicada en el IRQ-0001 ( <i>Usuario</i> ).
	3	El usuario introduce los datos correspondientes
	4	El sistema comprueba los datos.
	5	El usuario confirma el ingreso.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el sistema detecta algún error en la validez de los datos volverá al paso 2.
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene dos transacciones de intercambio de datos entre usuario y sistema.

<b>UC-0008</b>	<b>Logout usuario</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	
<b>Dependencias</b>	OBJ-0002 Gestionar información de usuario IRQ-0001 Usuario NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0007 Login usuario

<b>Descripción</b>	El sistema permitirá a un usuario registrado cerrar la conexión con las funcionalidades sólo disponibles para usuarios registrados.
--------------------	---

<b>Precondición</b>	Sólo puede realizarse tras haber realizado UC-0007 (Login usuario)	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de salir
	2	El usuario confirma al sistema que quiere cerrar su sesión
	3	El sistema elimina datos de sesión
	4	El sistema redirige al usuario a la zona de inicio de sesión de la aplicación
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene una sola transacción donde el usuario trata de salir de la aplicación y el sistema destruye la sesión.

<b>3 UC-0009</b>	<b>Modificar usuario</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	
<b>Dependencias</b>	OBJ-0002 Gestionar información de usuario IRQ-0001 Usuario NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0007 Login usuario
<b>Descripción</b>	El sistema debe permitir la modificación automática de los datos asociados a la configuración de usuario.
<b>Precondición</b>	Debe existir dicha información

	<b>Paso</b>	<b>Acción</b>
--	-------------	---------------

<b>Secuencia normal</b>	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de modificar datos de usuario
	2	El sistema recupera la información del usuario, detallada en el IRQ-0001 ( <i>Usuario</i> ).
	3	El usuario introduce los nuevos datos
	4	El sistema guarda la nueva información
	5	El sistema notifica, a través de un aviso general, del cambio realizado sobre las preferencias
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si alguno de los datos introducidos no es válido, se le notificará al usuario y se le pedirá que lo modifique.
	4	Si se produce algún error al actualizar los datos se le notifica al usuario y el caso de uso terminará.
<b>Estabilidad</b>		
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene dos transacciones, la primera en la que el usuario selecciona la opción de modificar sus datos y el sistema recupera sus datos. La segunda donde el usuario introduce los nuevos datos y el sistema los almacena.

<b>UC-0010</b>	<b>Mandar correo</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	

<b>Dependencias</b>	OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos OBJ-0004 Localizar puntos de interés IRQ-0001 Usuario IRQ-0002 Reto NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma	
<b>Descripción</b>	El sistema debe permitir el envío de correo indicando posibles problemas	
<b>Precondición</b>	Ninguna	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de enviar correo.
	2	El sistema pide al usuario el destinatario, asunto y cuerpo del correo
	3	El usuario indica los datos pedidos y confirma el envío
	4	El sistema valida los datos
5	El sistema realiza el envío y notifica al usuario	
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

	4	Si el sistema detecta algún error en la validez de los datos volverá al paso 2.
	5	Si se produce algún error el sistema informa al usuario y termina el caso de uso
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene dos transacciones, la primera en la que el usuario selecciona la opción de enviar un correo y el sistema le solicita los datos requeridos. La segunda donde el usuario introduce los datos, el sistema los valida y envía el mensaje.

<b>UC-0011</b>	<b>Consultar puntuaciones</b>
<b>Versión</b>	1.0 (01/03/2022)
<b>Autores</b>	Miguel Cabezas Puerto
<b>Fuentes</b>	
Dependencias	OBJ-0001 Gestionar puntuaciones IRQ-0001 Usuario IRQ-0002 Reto NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma UC-0007 Login usuario

<b>Descripción</b>	El sistema debe permitir el envío de correo indicando posibles problemas	
<b>Precondición</b>	Ninguna	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Un actor de tipo ACT-01 ( <i>Usuario</i> ) selecciona la opción de ver clasificaciones y puntuaciones.
	2	El sistema busca las puntuaciones de todos los usuarios y las ordena
	3	El sistema devuelve las puntuaciones ordenadas al usuario
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si se produce algún error el sistema informa al usuario y termina el caso de uso
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

- ◇ Complejidad: simple ya que contiene una única transacción donde el usuario solicita ver la clasificación y el sistema la calcula y muestra

### 3.2 Determinar la complejidad de los factores técnicos y del entorno

#### 2.3.1 Complejidad factores técnicos

Se entiende por complejidad de los factores técnicos, la influencia de cada factor en el sistema. Su valor se encuentra entre 0 (no tiene influencia) a 5 (esencial).

- ◇ Sistemas distribuidos: '4' se requiere la conexión con servidores externos y dicha interacción sucede continuamente al recuperar datos de remoto.
- ◇ Rendimiento: '3' ya que un error en la información tratada supone un perjuicio para el usuario, pero no de extrema gravedad ya que puede corroborar la información más adelante.
- ◇ Eficiencia del usuario final: '4' ya que el sistema está orientado a ayudar y facilitar al usuario la búsqueda de puntos de interés en un entorno competitivo.
- ◇ Procesamiento interno complejo: '3' ya que la gestión interna de los datos es sencilla al tratarse de ingresos, borrados y búsquedas de información en bases de datos. No obstante, debido a que la aplicación debe hacer uso de APIs externas,

conexiones a Internet e interactuar con aplicaciones de terceros, la complejidad de procesamiento aumenta.

- ◇ Reusabilidad: ‘2’ debido a que el código pretende realizar únicamente las funciones para las que está diseñado con leves ampliaciones solo para sistemas Android.
- ◇ Facilidad de instalación: ‘5’ ya que solo será necesario que el usuario pulse sobre el archivo APK en su dispositivo móvil para poder instalar la aplicación y este se genera automáticamente con la herramienta Android Studio
- ◇ Facilidad de uso: ‘5’ debido a que el sistema dispone de una interfaz y el procesamiento complejo es transparente al usuario.
- ◇ Portabilidad: ‘3’ debido a que pese a que el software está desarrollado solo para un sistema operativo Android, se puede utilizar en la mayoría de versiones.
- ◇ Facilidad de cambio: ‘4’ ya que el software está diseñado para que cualquier cambio o ampliación no repercute en las demás funcionalidades.
- ◇ Concurrencia: ‘2’ ya que no debe procesar datos en el mismo intervalo de tiempo ni realizar múltiples operaciones simultáneamente ya que esta parte se delega en la plataforma de Google.
- ◇ Características especiales de seguridad: ‘3’ ya que aunque los datos no requieren de ningún tipo de confidencialidad al poder ser inventados por el usuario conector de ser visibles al resto de jugadores, se dispone de contraseñas y sistemas de autenticación más sensibles.
- ◇ Acceso directo a terceras partes: ‘3’ ya que la aplicación debe interactuar con servidores y otros programas externos.
- ◇ Se requiere entrenamiento especial del usuario: ‘2’ ya que cualquier usuario que disponga de un dispositivo Android sin ningún otro conocimiento podrá utilizar la aplicación.

### 2.3.2 Complejidad factores del entorno

Se entiende por complejidad de los factores del entorno, la influencia de cada factor en el sistema. Su valor se encuentra entre 0 (no tiene influencia) a 5 (esencial).

- ◇ Familiaridad con UML: ‘2’ debido a que el equipo de desarrollo está formado por un integrante que ha cursado las asignaturas de Ingeniería del Software I y II y Gestión de Proyectos. Pero no es experto en UML.
- ◇ Trabajadores a tiempo parcial: ‘5’ debido a que el desarrollador se encuentra trabajando en una empresa mientras realiza el máster.

- ◇ Capacidad de los analistas: ‘3’ debido a que el integrante del equipo sigue formándose.
- ◇ Experiencia en la aplicación: ‘3’ ya que el desarrollador tiene conocimientos previos en la programación Android, pero no demasiada experiencia.
- ◇ Experiencia en orientación a objetos: ‘3’ debido a que el integrante del equipo tiene nociones de lenguajes orientados a objetos y experiencia laboral pero tampoco muy prolongada.
- ◇ Motivación: ‘3’ debido a que el proyecto al tratarse del asociado al fin de máster ilusiona al desarrollador. Por otro lado, siente la presión de verse sobrepasado ya que el carácter del proyecto difiere de las temáticas tratadas hasta ahora.
- ◇ Dificultad del lenguaje de programación: ‘3’ debido a que se trata del lenguaje JAVA, orientado a objetos, que, como mencionamos anteriormente, es un campo (la orientación a objetos), donde el desarrollador sigue formándose.
- ◇ Estabilidad de los requisitos: ‘4’ ya que, pese a pequeñas variaciones e interpretaciones, los aspectos requeridos no cambian ya que se encuentran recogidos en una propuesta del propio alumno aprobada por los órganos competentes de la facultad de Ciencias.

### 3.3 Conclusión

Una vez analizados los factores y casos de uso y habiéndoles asignado su respectiva complejidad, se introducen los datos en la herramienta EZEstimate para que realice los cálculos necesarios para obtener una predicción del nº de horas que conllevará el desarrollo del proyecto. Cabe destacar que se han bajado las horas por caso de uso de 20 (las establecidas por defecto), a 15 horas debido a que estos tienen un grado de refinamiento superior al habitual.

Se obtiene un nº total de 1010,88 horas, para un solo trabajador, es decir, tomando que el desarrollador trabaja de media 5 horas al día cada día de la semana natural, son 202,176 días naturales, que transformado a semanas resultan 28 semanas lo que equivale, contando 4 semanas por mes, 7 meses. Todo ello contando que no se toma la semana laboral sino la natural ni festivos ya que el estudiante podrá invertir horas en el proyecto sin seguir ningún convenio laboral.

Comenzando en marzo un proceso de análisis y toma de requisitos, el proyecto terminaría en septiembre con tiempo suficiente para cumplir los plazos de entrega marcados por la Facultad de Ciencias.

EZEstimate - D:\MasterIngenieriaInformatica\TFM\Documentacion\EZEstimation.ezp

File Settings Help

**Module**

Project

Add Module Delete

**Summary**

Total Modules: 1 **Excel Report** Generate Report

Use cases: Simple 11 Average 0 Complex 0

Actors: Simple 0 Average 1 Complex 1

**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: 5

UUCW: 55

UUPC = UAW + UUCW: 60

TFactor: 48

EFactor: 12

TCF = 0.6 + (.01\*TFactor): 1,08

EF = 1.4 + (-0.03\*EFactor): 1,04

UCP = UUPC\*TCT\*EF: 67,392

Total Effort@ 15 Hrs/UCP: 1010,88

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

Id	Module	Type	Name	complexity
1	Project	Actor	Usuario	Complex
10	Project	Usecase	Logout usuario	Simple
11	Project	Usecase	Modificar usuario	Simple
12	Project	Usecase	Mandar correo	Simple
13	Project	Usecase	Consultar puntu...	Simple
2	Project	Actor	Sistema	Average
3	Project	Usecase	Guardar tiempo ...	Simple
4	Project	Usecase	Iniciar reto	Simple
5	Project	Usecase	Mostrar reto en ...	Simple
6	Project	Usecase	Detener reto	Simple
7	Project	Usecase	Escanear QR	Simple
8	Project	Usecase	Registrar usuario	Simple
9	Project	Usecase	Login usuario	Simple

Ilustración 1 Estimación esfuerzo EZEstimate

### 3. Planificación temporal

Este apartado recoge la planificación temporal que se ha seguido en el desarrollo del proyecto. Con este objetivo, se mostrará su diagrama de Gantt, que muestra la distribución de tiempo para cada una de las tareas que se han llevado a cabo. Se mostrarán también otros datos de interés como el calendario de trabajo seguido y la distribución y uso de los recursos humanos.

Un diagrama de Gantt es una representación gráfica de las actividades sobre una escala de tiempos. Las actividades se representan en forma de barra sobre dicha escala manteniendo la relación de proporcionalidad entre sus duraciones y su representación gráfica, y su posición respecto del punto origen del proyecto. No permite la representación de conexiones cruzadas que muestre directamente la dependencia de tareas ni conocer claramente la lógica utilizada en la planificación.

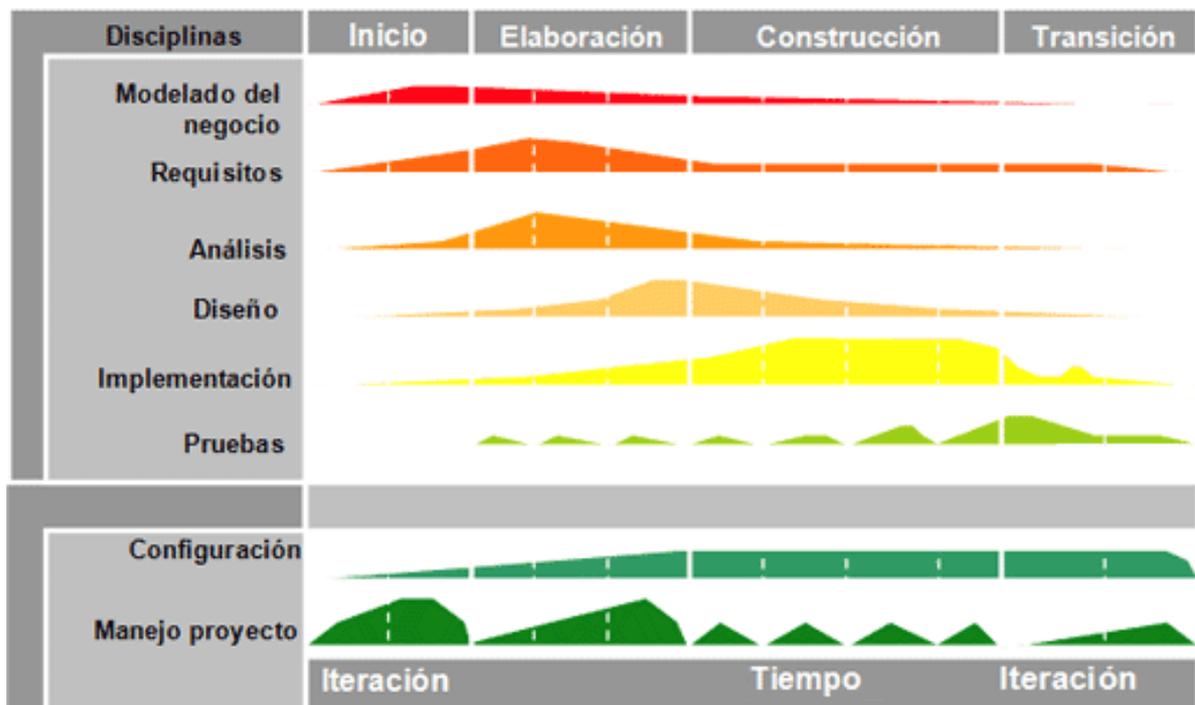
El marco de trabajo seguido en el desarrollo del proyecto se trata del Proceso Unificado cuyas características principales son:

- ◇ Conducido por casos de uso

- ◇ Centrado en la arquitectura
- ◇ Iterativo e incremental

Esto queda reflejado en el diagrama de Gantt, donde podemos ver cómo se solapan las actividades y se realizan diferentes iteraciones.

En la siguiente imagen podemos ver cómo se distribuye, de manera teórica en el proceso unificado, la carga de trabajo de cada tarea en relación con cada una de las fases fundamentales y cómo se solapan las tareas. El número de iteraciones en cada fase es modificable, pero deberá obedecer el esquema mostrado.



*Ilustración 2 Fases Proceso Unificado*

### 3.1 Calendario de trabajo

El proyecto comienza el 1 de marzo y tiene una duración de aproximadamente 7 meses, un total de 966,02 horas. Debido a que no se trata de un proyecto laboral, no se siguen convenios laborales en cuanto jornada laboral ni días no laborales. Se ha trabajado cuatro horas al día de media, decretando como días de no realización del proyecto los marcados en el siguiente calendario. Estos horarios son debidos a que el desarrollador del proyecto debe compatibilizarlo con su trabajo como desarrollador software.

### 3.2 Esquema de planificación temporal

La planificación temporal se utiliza para realizar la distribución del esfuerzo estimado en apartados anteriores. Para ello hay que identificar las tareas, así como asignarles tiempos y recursos, todo ello sin olvidarse de planificar la secuencia de ejecución de dichas tareas para

minimizar el tiempo de desarrollo del proyecto. El siguiente esquema muestra todas las tareas identificadas divididas a su vez en subtarefas con sus fechas de inicio y fin. También se indican los hitos que marcan el fin de cada iteración y cada fase.

▣ Inicio	2,04 días	mar 01/03/22	vie 04/03/22		
▣ Iteración 1	0,63 días	mar 01/03/22	mié 02/03/22		
▣ Modelado del negocio	0,5 días	mar 01/03/22	mar 01/03/22		
Reunión inicial tutor	3,98 horas	mar 01/03/22	mar 01/03/22		Miguel[50%]
Análisis aplicaciones similares	3 horas	mar 01/03/22	mar 01/03/22	4CC	Miguel[50%]
▣ Requisitos	0,63 días	mar 01/03/22	mié 02/03/22		
Identificar actores	1 hora	mar 01/03/22	mar 01/03/22	4	Miguel
Identificar y documentar objetivos generales	2 horas	mar 01/03/22	mar 01/03/22	7	Miguel
Identificar y documentar RNF básicos	2 horas	mar 01/03/22	mié 02/03/22	8	Miguel
Fin iteración 1	0 días	mié 02/03/22	mié 02/03/22	9	
▣ Iteración 2	1,42 días	mié 02/03/22	vie 04/03/22		
▣ Modelado de negocio	0,54 días	mié 02/03/22	mié 02/03/22		
Definición de calendario de trabajo	0,33 horas	mié 02/03/22	mié 02/03/22	10	Miguel
Determinación áreas a tratar	3 horas	mié 02/03/22	mié 02/03/22	13	Miguel[50%]
Búsqueda y experimentación herramientas auxiliares y apps similares	4 horas	mié 02/03/22	mié 02/03/22	13CC	Miguel[50%]
▣ Requisitos	0,75 días	mié 02/03/22	jue 03/03/22		
Identificar y documentar objetivos específicos	4 horas	mié 02/03/22	jue 03/03/22	14	Miguel[50%]
Identificación RI básicos	2 horas	jue 03/03/22	jue 03/03/22	17	Miguel
Refinamiento RNF	3 horas	mié 02/03/22	jue 03/03/22	17CC	Miguel[50%]
▣ Análisis	0,25 días	jue 03/03/22	vie 04/03/22		

Elaboración propuesta arquitectura básica	2 horas	jue 03/03/22	vie 04/03/22	18	Miguel
Fin iteración 2	0 días	vie 04/03/22	vie 04/03/22	21	
Fin inicio	0 días	vie 04/03/22	vie 04/03/22	22	
▣ Elaboración	19,63 días	vie 04/03/22	jue 31/03/22		
▣ Iteración 1	5,63 días	vie 04/03/22	vie 11/03/22		
▣ Modelado de negocio	0,75 días	vie 04/03/22	sáb 05/03/22		
Reunión tutor y relectura de restricciones para información detallada	6 horas	vie 04/03/22	sáb 05/03/22	23	Miguel
▣ Requisitos	0,63 días	sáb 05/03/22	sáb 05/03/22		
Identificar y documentar CU	4 horas	sáb 05/03/22	sáb 05/03/22	27	Miguel[50%]
Identificar y documentar objetivos operativos	4 horas	sáb 05/03/22	sáb 05/03/22	29CC	Miguel[50%]
Identificar restricciones	1 hora	sáb 05/03/22	sáb 05/03/22	30	Miguel
▣ Análisis	3,75 días	sáb 05/03/22	vie 11/03/22		
Desarrollo diagrama de clases	20 horas	sáb 05/03/22	mié 09/03/22	31	Miguel[25%]
▣ Realización CU subsistema 1	3,75 días	sáb 05/03/22	vie 11/03/22		
Realización CU retos	30 horas	sáb 05/03/22	vie 11/03/22	33CC	Miguel[50%]
▣ Diseño	0,5 días	vie 11/03/22	vie 11/03/22		
Estructura a seguir subsistema de diseño	4 horas	vie 11/03/22	vie 11/03/22	35	Miguel[50%]

<b>4 Implementación</b>	<b>0,25 días</b>	<b>sáb 05/03/22</b>	<b>dom 06/03/22</b>		
Determinar características huésped óptimo del sistema	2 horas	sáb 05/03/22	dom 06/03/22	33CC	Miguel[25%]
<b>4 Pruebas</b>	<b>0,5 días</b>	<b>vie 11/03/22</b>	<b>vie 11/03/22</b>		
Rendimiento y funcionalidad de aplicaciones similares en Android	4 horas	vie 11/03/22	vie 11/03/22	37CC	Miguel[50%]
Fin iteración 1	0 días	vie 11/03/22	vie 11/03/22	41	
<b>4 Iteración 2</b>	<b>6 días</b>	<b>vie 11/03/22</b>	<b>dom 20/03/22</b>		
<b>4 Modelado de negocio</b>	<b>0,25 días</b>	<b>vie 11/03/22</b>	<b>sáb 12/03/22</b>		
Revisión funcionalidades concretas aplicación	2 horas	vie 11/03/22	sáb 12/03/22	42	Miguel
<b>4 Requisitos</b>	<b>0,38 días</b>	<b>sáb 12/03/22</b>	<b>sáb 12/03/22</b>		
Refinamiento RI	3 horas	sáb 12/03/22	sáb 12/03/22	45	Miguel
<b>4 Análisis</b>	<b>4,38 días</b>	<b>sáb 12/03/22</b>	<b>vie 18/03/22</b>		
Documentar clases (glosario)	5 horas	sáb 12/03/22	dom 13/03/22	47	Miguel[25%]
<b>4 Realización CU Subsistema 2</b>	<b>4,38 días</b>	<b>sáb 12/03/22</b>	<b>vie 18/03/22</b>		
Realización CU Usuarios	35 horas	sáb 12/03/22	vie 18/03/22	49CC	Miguel[35%]
<b>4 Diseño</b>	<b>0,5 días</b>	<b>vie 18/03/22</b>	<b>sáb 19/03/22</b>		
Refinamiento subsistema de diseño	4 horas	vie 18/03/22	sáb 19/03/22	51	Miguel
<b>4 Implementación</b>	<b>2,5 días</b>	<b>sáb 12/03/22</b>	<b>mié 16/03/22</b>		
Implementación pequeñas aplicaciones Android	20 horas	sáb 12/03/22	mié 16/03/22	51CC	Miguel[35%]
<b>4 Pruebas</b>	<b>0,5 días</b>	<b>sáb 19/03/22</b>	<b>dom 20/03/22</b>		
Rendimiento y funcionalidad apps	4 horas	sáb 19/03/22	dom 20/03/22	53	Miguel
Rendimiento y funcionalidad apps similares en huésped óptimo	4 horas	sáb 19/03/22	dom 20/03/22	53	Miguel
Fin iteración 2	0 días	dom 20/03/22	dom 20/03/22	57	
<b>4 Iteración 3</b>	<b>8 días</b>	<b>dom 20/03/22</b>	<b>jue 31/03/22</b>		
<b>4 Modelado de negocio</b>	<b>0,13 días</b>	<b>dom 20/03/22</b>	<b>dom 20/03/22</b>		
Dudas a tutor previas a la construcción	1 hora	dom 20/03/22	dom 20/03/22	58	Miguel
<b>4 Requisitos</b>	<b>0,38 días</b>	<b>dom 20/03/22</b>	<b>dom 20/03/22</b>		
Refinamiento de objetivos operativos	2 horas	dom 20/03/22	dom 20/03/22	61	Miguel
Realizar matrices de rastreabilidad	1 hora	dom 20/03/22	dom 20/03/22	63	Miguel
<b>4 Análisis</b>	<b>2,5 días</b>	<b>dom 20/03/22</b>	<b>jue 24/03/22</b>		
Refinamiento realización CU subsistema 1	15 horas	dom 20/03/22	mié 23/03/22	64	Miguel[50%]
Refinamiento realización CU subsistema 2	10 horas	dom 20/03/22	mar 22/03/22	66CC	Miguel[50%]
<b>4 Realización CU Subsistema 3</b>	<b>0,63 días</b>	<b>mié 23/03/22</b>	<b>jue 24/03/22</b>		
Realización CU Preferencias usuario	5 horas	mié 23/03/22	jue 24/03/22	66	Miguel[50%]
Realización CU Ayuda	3 horas	mié 23/03/22	jue 24/03/22	69CC	Miguel[50%]
<b>4 Diseño</b>	<b>1 día</b>	<b>jue 24/03/22</b>	<b>vie 25/03/22</b>		
Realización clases diseño subsistema 1	4 horas	jue 24/03/22	vie 25/03/22	69	Miguel

Realización clases diseño subsistema 2	4 horas	vie 25/03/22	vie 25/03/22	72	Miguel[75%]
Realización clases diseño subsistema 3	2 horas	vie 25/03/22	vie 25/03/22	73CC	Miguel[25%]
<b>4 Implementación</b>	<b>3,5 días</b>	<b>vie 25/03/22</b>	<b>mié 30/03/22</b>		
Pequeñas app en Android similares	24 horas	vie 25/03/22	mié 30/03/22	73	Miguel
Determinar lógica de comunicación	2 horas	mié 30/03/22	mié 30/03/22	76	Miguel
Determinar BD a utilizar	2 horas	mié 30/03/22	mié 30/03/22	77	Miguel
<b>4 Pruebas</b>	<b>0,5 días</b>	<b>mié 30/03/22</b>	<b>jue 31/03/22</b>		
Pruebas unitarias BD en sistema operativo	4 horas	mié 30/03/22	jue 31/03/22	78	Miguel
Fin iteración 3	0 días	jue 31/03/22	jue 31/03/22	80	
Fin elaboración	0 días	jue 31/03/22	jue 31/03/22	81	
<b>4 Construcción</b>	<b>94,63 días</b>	<b>jue 31/03/22</b>	<b>jue 11/08/22</b>		
<b>4 Iteración 1</b>	<b>22,75 días</b>	<b>jue 31/03/22</b>	<b>lun 02/05/22</b>		
<b>4 Modelado de negocio</b>	<b>0,13 días</b>	<b>jue 31/03/22</b>	<b>jue 31/03/22</b>		
Reunión tutor comprobación todos requisitos	1 hora	jue 31/03/22	jue 31/03/22	82	Miguel
<b>4 Requisitos</b>	<b>0,25 días</b>	<b>jue 31/03/22</b>	<b>vie 01/04/22</b>		
Validación de todos requisitos	2 horas	jue 31/03/22	vie 01/04/22	86	Miguel
<b>4 Análisis</b>	<b>0,63 días</b>	<b>vie 01/04/22</b>	<b>vie 01/04/22</b>		
Refinamiento realización CU Subsistema 3	5 horas	vie 01/04/22	vie 01/04/22	88	Miguel[50%]

<b>4 Diseño</b>	<b>0,88 días</b>	<b>vie 01/04/22</b>	<b>sáb 02/04/22</b>		
Refinamiento clases diseño subsistema 1	3 horas	vie 01/04/22	vie 01/04/22	90CC	Miguel[50%]
Refinamiento clases diseño subsistema 2	3 horas	vie 01/04/22	sáb 02/04/22	92	Miguel[50%]
Refinamiento clases diseño subsistema 3	2 horas	vie 01/04/22	sáb 02/04/22	90	Miguel[50%]
<b>4 Implementación</b>	<b>20 días</b>	<b>sáb 02/04/22</b>	<b>sáb 30/04/22</b>		
Implementación componentes separados subsistema 1	110 horas	sáb 02/04/22	jue 21/04/22	94	Miguel
Implementación BD para cada componente de subsistema 1	50 horas	jue 21/04/22	sáb 30/04/22	96	Miguel[60%]
<b>4 Pruebas</b>	<b>7,75 días</b>	<b>jue 21/04/22</b>	<b>lun 02/05/22</b>		
Pruebas unitarias cada componente en local subsistema 1	6 horas	jue 21/04/22	vie 22/04/22	96	Miguel[40%]
Pruebas unitarias cada componente Firebase subsistema 1	12 horas	sáb 30/04/22	lun 02/05/22	97	Miguel
Fin iteración 1	0 días	lun 02/05/22	lun 02/05/22	100	
<b>4 Iteración 2</b>	<b>20,25 días</b>	<b>lun 02/05/22</b>	<b>lun 30/05/22</b>		
<b>4 Modelado de negocio</b>	<b>0,13 días</b>	<b>lun 02/05/22</b>	<b>lun 02/05/22</b>		
Log pruebas iteración anterior	1 hora	lun 02/05/22	lun 02/05/22	101	Miguel
<b>4 Requisitos</b>	<b>0,13 días</b>	<b>lun 02/05/22</b>	<b>lun 02/05/22</b>		
Validar que los requisitos no han cambiado	1 hora	lun 02/05/22	lun 02/05/22	104	Miguel

▣ <b>Análisis</b>	<b>0,75 días</b>	<b>lun 02/05/22</b>	<b>mar 03/05/22</b>			
Unificar realización CU todos subsistemas	6 horas	lun 02/05/22	mar 03/05/22	106	Miguel[50%]	
▣ <b>Diseño</b>	<b>0,75 días</b>	<b>lun 02/05/22</b>	<b>mar 03/05/22</b>			
Unificación clases diseño subsistema 1	6 horas	lun 02/05/22	mar 03/05/22	108CC	Miguel[25%]	
▣ <b>Implementación</b>	<b>17,5 días</b>	<b>lun 02/05/22</b>	<b>vie 27/05/22</b>			
Integrar componentes subsistema 1 en subsistema 1	80 horas	mar 03/05/22	mié 18/05/22	110	Miguel[25%]	
Implementación componentes separados subsistema 2	120 horas	lun 02/05/22	lun 23/05/22	110CC	Miguel[25%]	
Implementación BD para cada componente subsistema 2	20 horas	lun 23/05/22	vie 27/05/22	113	Miguel[75%]	
▣ <b>Pruebas</b>	<b>9,25 días</b>	<b>mié 18/05/22</b>	<b>lun 30/05/22</b>			
Pruebas subsistema 1 integrado con Firebase	30 horas	mié 18/05/22	lun 23/05/22	112	Miguel[75%]	
Pruebas unitarias cada componente local subsistema 2	10 horas	lun 23/05/22	mié 25/05/22	113	Miguel[25%]	
Pruebas unitarias cada componente Firebase subsistema 2	20 horas	vie 27/05/22	lun 30/05/22	114	Miguel	
Fin iteración 2	0 días	lun 30/05/22	lun 30/05/22	118		
▣ <b>Iteración 3</b>	<b>24,63 días</b>	<b>lun 30/05/22</b>	<b>lun 04/07/22</b>			
▣ <b>Modelado de negocio</b>	<b>0,13 días</b>	<b>lun 30/05/22</b>	<b>lun 30/05/22</b>			
Log pruebas iteración anterior	1 hora	lun 30/05/22	lun 30/05/22	119	Miguel	
▣ <b>Requisitos</b>	<b>0,13 días</b>	<b>lun 30/05/22</b>	<b>mar 31/05/22</b>			
Validar que los requisitos no han cambiado	1 hora	lun 30/05/22	mar 31/05/22	122	Miguel	
▣ <b>Análisis</b>	<b>0,5 días</b>	<b>mar 31/05/22</b>	<b>mié 01/06/22</b>			
Refinamiento unificación realización CU todos subsistemas	4 horas	mar 31/05/22	mié 01/06/22	124	Miguel[25%]	
▣ <b>Diseño</b>	<b>0,5 días</b>	<b>mar 31/05/22</b>	<b>mié 01/06/22</b>			
Unificación clases de diseño subsistema 2	4 horas	mar 31/05/22	mié 01/06/22	126CC	Miguel[25%]	
▣ <b>Implementación</b>	<b>21,25 días</b>	<b>mar 31/05/22</b>	<b>jue 30/06/22</b>			
Integrar componentes subsistema 2 en subsistema 2	80 horas	mié 01/06/22	mié 15/06/22	128	Miguel[50%]	
Implementación componentes separados subsistema 3	100 horas	mar 31/05/22	vie 17/06/22	128CC	Miguel[50%]	
Implementación almacenaje de datos para componentes subsistema 3	70 horas	vie 17/06/22	jue 30/06/22	131	Miguel	
▣ <b>Pruebas</b>	<b>13,88 días</b>	<b>mié 15/06/22</b>	<b>lun 04/07/22</b>			
Pruebas subsistema 2 integrado con Firebase	12 horas	mié 15/06/22	vie 17/06/22	130	Miguel[50%]	
Pruebas almacenaje componentes subsistema 3	25 horas	jue 30/06/22	lun 04/07/22	132	Miguel	
Fin iteración 3	0 días	lun 04/07/22	lun 04/07/22	135		
▣ <b>Iteración 4</b>	<b>14,13 días</b>	<b>lun 04/07/22</b>	<b>dom 24/07/22</b>			
▣ <b>Modelado de negocio</b>	<b>0,13 días</b>	<b>lun 04/07/22</b>	<b>lun 04/07/22</b>			
Log pruebas iteración anterior	1 hora	lun 04/07/22	lun 04/07/22	136	Miguel	
▣ <b>Requisitos</b>	<b>0,13 días</b>	<b>lun 04/07/22</b>	<b>lun 04/07/22</b>			
Validar que los requisitos no han cambiado	1 hora	lun 04/07/22	lun 04/07/22	139	Miguel	
▣ <b>Análisis</b>	<b>0,13 días</b>	<b>lun 04/07/22</b>	<b>lun 04/07/22</b>			
Validación unificación realización CU todos subsistemas	1 hora	lun 04/07/22	lun 04/07/22	141	Miguel	

✦ <b>Diseño</b>	<b>0,38 días</b>	<b>lun 04/07/22</b>	<b>mar 05/07/22</b>		
Unificación clases de diseño subsistema 3	3 horas	lun 04/07/22	mar 05/07/22	143	Miguel[50%]
✦ <b>Implementación</b>	<b>11,25 días</b>	<b>lun 04/07/22</b>	<b>mié 20/07/22</b>		
Integrar componentes subsistema 3 en subsistema 3	40 horas	mar 05/07/22	mar 12/07/22	145	Miguel[50%]
Integrar subsistema 1 y subsistema 2	90 horas	lun 04/07/22	mié 20/07/22	145CC	Miguel[50%]
✦ <b>Pruebas</b>	<b>2,5 días</b>	<b>mié 20/07/22</b>	<b>dom 24/07/22</b>		
Pruebas subsistema 1 y subsistema 2 integrados en SQL	20 horas	mié 20/07/22	dom 24/07/22	148	Miguel
Fin iteración 4	0 días	dom 24/07/22	dom 24/07/22	150	
✦ <b>Iteración 5</b>	<b>12,88 días</b>	<b>dom 24/07/22</b>	<b>jue 11/08/22</b>		
✦ <b>Modelado de negocio</b>	<b>0,38 días</b>	<b>dom 24/07/22</b>	<b>dom 24/07/22</b>		
Log pruebas iteración anterior	1 hora	dom 24/07/22	dom 24/07/22	151	Miguel[25%]
Estudio APIs geolocalización	3 horas	dom 24/07/22	dom 24/07/22	154CC	Miguel[75%]
✦ <b>Requisitos</b>	<b>0,13 días</b>	<b>dom 24/07/22</b>	<b>dom 24/07/22</b>		
Validar que los requisitos no han cambiado	1 hora	dom 24/07/22	dom 24/07/22	155	Miguel
✦ <b>Diseño</b>	<b>0,5 días</b>	<b>dom 24/07/22</b>	<b>lun 25/07/22</b>		
Unificación clases de diseño subsistema 1,2, y 3	4 horas	dom 24/07/22	lun 25/07/22	157	Miguel
✦ <b>Implementación</b>	<b>8,75 días</b>	<b>lun 25/07/22</b>	<b>sáb 06/08/22</b>		
Integrar subsistemas 1,2 y 3	70 horas	lun 25/07/22	sáb 06/08/22	159	Miguel[60%]
Implementar código geolocalización en API elegida	20 horas	lun 25/07/22	vie 29/07/22	161CC	Miguel[40%]
✦ <b>Pruebas</b>	<b>9,38 días</b>	<b>vie 29/07/22</b>	<b>jue 11/08/22</b>		
Pruebas subsistema 1, subsistema 2 y	25 horas	sáb 06/08/22	jue 11/08/22	161	Miguel
Pruebas subsistema 1, subsistema 2 y subsistema 3 integrados sin geolocalización	25 horas	sáb 06/08/22	jue 11/08/22	161	Miguel
Pruebas unitarias geolocalización	4 horas	vie 29/07/22	vie 29/07/22	162	Miguel[40%]
Fin iteración 5	0 días	jue 11/08/22	jue 11/08/22	164	
Fin construcción	0 días	jue 11/08/22	jue 11/08/22	166	
✦ <b>Transición</b>	<b>24,13 días</b>	<b>jue 11/08/22</b>	<b>mié 14/09/22</b>		
✦ <b>Iteración 1</b>	<b>17,75 días</b>	<b>jue 11/08/22</b>	<b>lun 05/09/22</b>		
✦ <b>Modelado de negocio</b>	<b>0,13 días</b>	<b>jue 11/08/22</b>	<b>jue 11/08/22</b>		
Log pruebas construcción	1 hora	jue 11/08/22	jue 11/08/22	167	Miguel
✦ <b>Requisitos</b>	<b>0,13 días</b>	<b>jue 11/08/22</b>	<b>jue 11/08/22</b>		
Comprobar que los requisitos no han cambiado y validación requisitos geolocalización	1 hora	jue 11/08/22	jue 11/08/22	171	Miguel
✦ <b>Diseño</b>	<b>0,5 días</b>	<b>jue 11/08/22</b>	<b>vie 12/08/22</b>		
Refinamiento y validación de la unificación de las clases de diseño subsistemas 1,2,3	4 horas	jue 11/08/22	vie 12/08/22	173	Miguel[20%]
✦ <b>Implementación</b>	<b>11,25 días</b>	<b>jue 11/08/22</b>	<b>sáb 27/08/22</b>		
Integrar geolocalización con subsistemas	90 horas	jue 11/08/22	sáb 27/08/22	175CC	Miguel[80%]
✦ <b>Pruebas</b>	<b>6,25 días</b>	<b>sáb 27/08/22</b>	<b>lun 05/09/22</b>		
Pruebas subsistemas junto con geolocalización con usuarios	50 horas	sáb 27/08/22	lun 05/09/22	177	Miguel
Fin iteración 1	0 días	lun 05/09/22	lun 05/09/22	179	
✦ <b>Iteración 2</b>	<b>6,38 días</b>	<b>lun 05/09/22</b>	<b>mié 14/09/22</b>		
✦ <b>Modelado de negocio</b>	<b>0,13 días</b>	<b>lun 05/09/22</b>	<b>lun 05/09/22</b>		
Reunión tutor presentación	1 hora	lun 05/09/22	lun 05/09/22	180	Miguel

▣ <b>Requisitos</b>	<b>0,25 días</b>	<b>lun 05/09/22</b>	<b>lun 05/09/22</b>		
Refinamiento final	2 horas	lun 05/09/22	lun 05/09/22	183	Miguel[20%]
▣ <b>Implementación</b>	<b>2,5 días</b>	<b>lun 05/09/22</b>	<b>jue 08/09/22</b>		
Perfeccionamiento geolocalización	20 horas	lun 05/09/22	jue 08/09/22	185CC	Miguel[80%]
▣ <b>Pruebas</b>	<b>3,75 días</b>	<b>jue 08/09/22</b>	<b>mié 14/09/22</b>		
Pruebas finales con usuarios en huésped menos óptimo	30 horas	jue 08/09/22	mié 14/09/22	187	Miguel
Fin iteración 2	0 días	mié 14/09/22	mié 14/09/22	189	
Fin transición	0 días	mié 14/09/22	mié 14/09/22	190	Miguel

*Ilustración 3 Tareas*

### 3.2.1 Diagrama de Gantt

Se ha usado un diagrama de Gantt para representar gráficamente las dependencias entre tareas y subtareas indicadas en el apartado anterior. Para su visualización se recomienda acceder al archivo *PlanificacionTemporal.mpp*

### 3.3 Uso de recursos

Disponer de recursos y distribuirlos de la mejor forma para realizar las tareas para así conseguir una ocupación de estos lo más cercana al 100% resulta un aspecto fundamental a la hora de realizar una buena planificación temporal.

Dado que se trata de un proyecto individual, la paralelización de las tareas se torna más complicada, no obstante, en el diagrama de Gantt se puede observar cierta paralelización fundamentalmente en las tareas de análisis, diseño e implementación, llegando a coincidir algunas de las tareas de dichas fases obedeciendo el Proceso Unificado seguido.

A continuación, se muestra un diagrama con las horas asignadas al recurso.

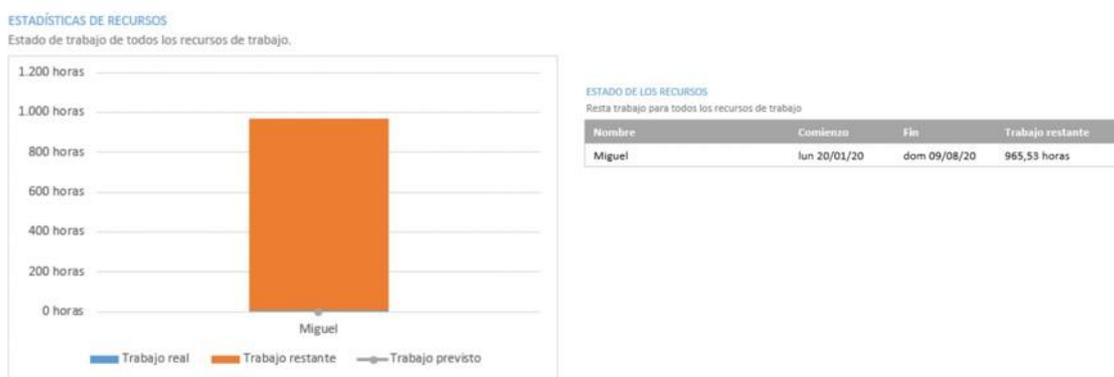


Ilustración 4 Uso de recursos