

## 0.1 Ejecución y control del proyecto

### 0.1.1 Reparto de trabajo y comunicación interna

Para la distribución de tareas y responsabilidades a lo largo del proyecto, se trató de buscar un consenso entre los miembros del grupo, de tal forma que cada uno pudiera desarrollar los componentes del sistema que prefiriera, de manera que durante las reuniones semanales se dedicaba un tiempo a asignar tareas entre los miembros. No obstante, para algunas de estas tareas como la redacción de diferentes secciones del plan de gestión, se decidió realizar la asignación de forma puramente aleatoria, basada en generadores de números aleatorios. En cuanto a la forma de establecer una forma de comunicación efectiva entre los integrantes del equipo, se disponía de un grupo de Whatsapp, así como de un servidor de la aplicación Discord. En el primero se discutían cuestiones que surgían durante el desarrollo más referentes al ámbito cotidiano, mientras que en el segundo se utilizaban los canales de voz (uno general, uno para backend, y dos para móvil y escritorio) habilitados para las reuniones semanales, donde se abarcaban tanto el control de esfuerzos, como la planificación de la siguiente semana. Con el paso de las semanas, debido a que a algunos miembros les funcionaba mejor Skype, se movieron las reuniones semanales a llamadas de esta plataforma, y Discord se seguía usando, más como comunicación casual durante el trabajo personal.

### 0.1.2 Control de progresos y de tareas pendientes

Para el control de los progresos de los miembros del equipo, se han mantenido reuniones semanales donde se monitorizaban las horas que cada miembro había invertido en el desarrollo, y se apuntan en una hoja de cálculo tipo Excel. Para mantener un control de las tareas realizadas y pendientes se utilizaban issues de Github, que se asignaban acorde al plan establecido en el diagrama de Gantt (sección 3.2.4), al calendario del proyecto y al progreso que se hubiera realizado, de forma que se procuraba no sobrecargar a los miembros, encontrando un compromiso con el plan inicial.

### 0.1.3 Adecuación de las herramientas y tecnologías empleadas

En un principio todos los integrantes tuvieron dificultades en el aprendizaje de las nuevas tecnologías a utilizar debido a la poca experiencia que se tenía con estas, no obstante después de que todos los componentes del equipo realizaran los procesos de formación mencionados en la sección 3.1 se pudo empezar a realizar las primeras implementaciones del producto final en su primera versión.

### 0.1.4 Ajustes realizados cuando se detectaron divergencias

Se ha procurado seguir el calendario propuesto en el Diagrama de Gantt de la sección 3.2.4 de este documento, sin embargo, no se ha conseguido realizar todas las actividades propuestas, puesto que se ha desestimado el tiempo para algunas de ellas, y la situación de emergencia sanitaria ha provocado que se dé una cierta desorganización durante las primeras semanas de confinamiento. Para corregir estos desajustes, los integrantes del grupo han dedicado un número extra de horas, especialmente en el período entre el miércoles 8 de Abril y miércoles 15 de abril.

### 0.1.5 Control de versiones del código, construcción y despliegue

Durante la implementación del código tanto para las versiones de escritorio como móvil del frontend y el modelo de aplicación del backend todo miembro del grupo tras realizar modificaciones en

el código hace los correspondientes commit y push en el repositorio de Github para que todo el personal involucrado contase con la versión mas reciente. Encaso de detectar conflictos en el momento de hacer push se comunicaba a los integrantes a los que afectaba la situación para ponerse de acuerdo sobre que modificación debía prevalecer para conseguir un resultado consistente. El despliegue para el backend se hizo creando un recurso web en Azure y poder poner a disposición del grupo la aplicación, para la base de datos nuevamente se creo otro recurso en Azure, en este caso un base de datos PostgreSQL con docker. En cuanto al despliegue en el frontend para la versión de escritorio se probaron los resultados de forma local en las máquinas del equipo, mismo modo de actuación que se siguió en la parte de móvil probando en los resultados en los simuladores que ponen a disposición los IDEs utilizados. Para la integración del backend se creo un nuevo repositorio de Github para poder desplegar el modelo implementado en python con Django, incluyendo un fichero **requirements.txt** en el que se incluían los paquetes necesarios y las versiones solventado así las dependencias, en cuanto a la conexión con la base de datos en el propio proyecto se configuró la conexión con esta. Principalmente no hubo problema en conectar la base de datos con el proyecto pero si al desplegar la aplicación, debido a que el navegador utilizado para las pruebas ,Mozilla en este caso, no accedía a una versión estable de la app al producirse fallos de compilación debido que no se incluían las dependencias en el **requirements.txt**. Durante el periodo de desarrollo transcurrido se han cometido ciertos errores por algún integrante del grupo de frontend conllevando a la pérdida parcial de contenido en el interfaz de escritorio, fallo que fue solucionado por el responsable tras darse cuenta de esto.

### 0.1.6 Pruebas de Software

En lo que respecta a las pruebas de software no se han implementado pruebas automáticas pero se ha probado de forma continuada y manual si todos los elementos del software cumplen correctamente el objetivo por el cual fueron implementados.