



VENERIS

Veneris

Ingeniería del Software I – Grupo Doble Grado de Ingeniería Informática y Matemáticas

Daniel Pulido Leo

Rodrigo Sánchez Peña

Alvaro Fu

David Tijerín Antón

Lucas Fernández-Galiano Catalán

Pablo Jofre Gómez

Lukas Otero Asumendi

ÍNDICE

1. Introducción.	3
2. Proceso scrum.	4
2.1. Estructura y funcionamiento del equipo scrum.	4
2.2. Historias de Usuario.	4
2.3. Sprint Reviews.	11
2.4. Sprint Retrospectives.	14
2.5. Sprint Plannings.	17
2.6. Product Backlog.	19
2.7. Sprint Backlog.	20
2.8. Descripción del contenido del repositorio en GitHub.	25
3. Otras tareas de gestión del proyecto.	28
3.1. Gestión de riesgos	28
3.2. RTF (Revisiones Técnicas Formales)	30
3.3. Revisión de la gestión de riesgos	35
4. Descripción del trabajo realizado por cada miembro del grupo.	37
5. Código.	44

1. Introducción.

En esta memoria se recogen todos los procesos realizados durante el cuatrimestre para realizar el proyecto Veneris, además de todo el material creado para que el producto final fuese el adecuado. Este material ha sido elaborado siguiendo los criterios de desarrollo scrum y por lo tanto los apartados de esta misma se dividirán en los elementos esenciales de un proceso de desarrollo scrum, además de todo aquello que se considere de relevancia para tener en cuenta a la hora de valorar el proyecto.

2. Proceso scrum.

2.1. Estructura y funcionamiento del equipo scrum.

En cuanto a la organización del equipo scrum, hemos considerado el escenario de que tanto el product owner como el scrum master eran externos a los siete integrantes del equipo, basándonos en la autoasignación de tareas y la comunicación para organizarnos de la mejor manera posible. En un inicio esto no funcionó como esperábamos, debido en gran parte a la falta de experiencia, lo que se tradujo en trabajo duplicado que acababa no siendo utilizado. Sin embargo, según iban avanzando los sprints hemos notado una mejora notable en todos estos aspectos. Para ello, hemos utilizado fuertemente los tableros kanban que proporcionan los proyectos de GitHub, así como la capacidad de crear tareas y subtareas y cambiar el estado de las mismas (todo, in progress, to test y Done). Además, GitHub permite asignar dichas tareas a uno o varios miembros del equipo en específico, lo que hemos utilizado para no solaparnos a la hora de desarrollar nuevas funcionalidades. Aparte de eso nos hemos apoyado en el grupo de Whatsapp para saber qué estaba haciendo cada uno en cada momento, y resolver dudas sobre el funcionamiento si surgían durante el desarrollo.

2.2. Historias de Usuario.

A continuación se presentan todas las historias de usuario planteadas para el proyecto, tanto las que se llevaron a cabo como las que se terminaron por descartar.

Se adjunta una descripción completa de cada historia, junto con sus condiciones de aceptación, su tamaño y su prioridad, además del estado de dicha historia y en qué sprint específico se desarrolló.

Cabe mencionar antes los distintos campos que hemos usado para el tamaño, prioridad y estado. Dentro del tamaño, hemos definido los campos siguientes: *XS*, *S*, *M*, *L*, *XL*, para describir cuán grande o pequeño es a nuestro parecer la historia de usuario a realizar.

La prioridad oscila entre *WH* (*won't have*), *CH* (*could have*), *SH* (*should have*), *MH* (*must have*), según cómo de necesitada e importante sea la historia para nuestro producto final.

Por otro lado, para el estado decidimos crear 4 campos: *Todo*, para aquellas tareas que seguían pendientes por hacer; *En Progreso*, para aquellas que habían sido empezadas por algún miembro; *Bajo Revisión*, para aquellas que, a priori, habían sido finalizadas, pero necesitaban ser revisadas por otro miembro que no haya participado en la realización de la tarea para maximizar la búsqueda de errores; y *Done*, el estado final, usado para aquellas tareas que han sido terminadas y revisadas correctamente.

(Implementadas)

- Como usuario nuevo, quiero crear un perfil para controlar todos mis datos tales como foto, nombre, hábitos o intereses.
 - Descripción: Cada usuario podrá crear su perfil público en el que podrá describir su personalidad, hábitos, gustos e intereses románticos, además de incluir una foto suya.

- C. de aceptación:
 - Opción de registrarse en la aplicación.
 - Preguntas al crear el usuario de:
 - Nombre.
 - Foto.
 - Hábitos.
 - Intereses románticos.
 - Personalidad.
 - Opción de modificar dichos datos una vez ya está la sesión iniciada.
 - Tamaño: L
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 2
 - Final: Sprint 4
- Como usuario, quiero ver perfiles de otros usuarios.
 - Descripción: La aplicación tendrá una sección donde el usuario podrá ver perfiles de otros, mostrándose la foto de perfil de los mismos. Al seleccionar un perfil completo, se enseñará el resto de datos de dicho usuario, y la opción de invitarle a jugar.
 - C. de aceptación:
 - Sección modo scroll down donde vayan apareciendo las fotos de perfil de distintos usuarios.
 - Opción de hacer click en cualquier foto y que se muestren los datos de dicho usuario
 - Opción de invitar a jugar a un usuario concreto
 - Tamaño: L
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 2
 - Final: Sprint 4
 - Como usuario, quiero que la web tenga servicios online para poder acceder desde cualquier ubicación.
 - Descripción: La página web estará subida a un servidor, que tendrá un dominio con el que se puede acceder desde cualquier navegador web.
 - C. de aceptación:
 - Accesible desde cualquier navegador moderno utilizando el dominio asignado.
 - Tamaño: L
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 3
 - Final: Sprint 3
 - Como usuario 1, quiero jugar a juegos cooperativos para conocer más al usuario 2.
 - Descripción: La aplicación de citas tendrá incorporada una sección para poder jugar a un juego cooperativo sencillo en línea con otros usuarios para ayudar al sistema a

medir la compatibilidad entre ambos, de forma que al completar juegos se añadan puntos a una pareja.

- C. de aceptación:
 - Opción de jugar a un juego cooperativo en la ventana de chat.
 - Acciones coordinadas entre ambos dispositivos.
 - Objetivo del juego definido y posibilidad de superar el nivel.
 - Sistema de puntos que mida la compatibilidad en base al número de niveles superados o partidas completadas.
 - Tamaño: XL
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 4
 - Final: Sprint 5
- Como usuario 1, quiero chatear con otros usuarios para poder conocernos mejor.
 - Descripción: Debe haber un sistema de comunicación entre distintos clientes vía chat mediante mensajes de texto con una interfaz para el chat con una persona específica. Dicha opción debe estar bloqueada detrás de un mínimo de puntos que se consiguen al avanzar lo suficiente en los distintos juegos cooperativos.
 - C. de aceptación:
 - Interfaz y comunicación vía mensajes entre distintos clientes.
 - Opción para permitir acceder a dicha funcionalidad solo cuando se ha conseguido un número mínimo de puntos.
 - Tamaño: M
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 3
 - Final: Sprint 3
 - Como usuario, quiero invitar a la gente para jugar juntos.
 - Descripción: Implementar un método que permita a los usuarios invitar a otros jugadores a participar en partidas conjuntas. La invitación se podría realizar desde el perfil del usuario
 - C. de aceptación:
 - Usuario puede enviar invitación directa a otro
 - Opción de invitar claramente accesible desde la interfaz
 - Producción de una notificación a la persona invitada
 - Registro de invitaciones aceptadas, rechazadas o ignoradas
 - Si se acepta la invitación, el juego se inicia automáticamente
 - Tamaño: M
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 4
 - Final: Sprint 4

- Como usuario, quiero que mi perfil tenga una contraseña para poder poner la cuenta en distintos aparatos.
 - Descripción: Implementación de un sistema de autenticación mediante contraseña para cada perfil. Así, el usuario puede abrir la cuenta en otros dispositivos sin perder información.
 - C. de aceptación:
 - El sistema permite crear una contraseña al abrir una cuenta
 - Inicio de sesión múltiple
 - Tamaño: S
 - Prioridad: MH
 - Estado: Done
 - Inicio: Sprint 2
 - Final: Sprint 6

- Como usuario, quiero variedad de juegos y de niveles para que no sea demasiado repetitivo al jugar con distintas personas.
 - Descripción: Añadir una variedad suficiente de juegos y niveles dentro de la app, asegurando así una experiencia de juego dinámica. Como objetivo inicial, se deben implementar los juegos del cuatro en raya y dos verdades y una mentira, aunque en un futuro se añadirán más.
 - C. de aceptación:
 - Juego implementado de dos verdades y una mentira
 - Juego implementado de cuatro en raya
 - Tamaño: XL
 - Prioridad: SH
 - Estado: Done

- Como usuario general, quiero poder definir mis intereses amorosos y buscar perfiles según dichos intereses para poder lograr una pareja adecuada para mí.
 - Descripción: Implementación de un sistema de búsqueda de perfiles con filtros personalizables para facilitar encontrar personas compatibles rápidamente. Los filtros deben ser en relación a los ajustes presentes en la personalización del usuario, por género y gustos.
 - C. de aceptación:
 - El usuario puede seleccionar sus intereses
 - El sistema usa los intereses para sugerir perfiles compatibles
 - La interfaz de edición respeta la estética de la app
 - La interfaz es clara y accesible desde el menú de inicio.
 - Actualización dinámica según los filtros seleccionados.
 - Tamaño: L
 - Prioridad: SH
 - Estado: Done
 - Inicio: Sprint 5
 - Final: Sprint 5

- Como usuario, quiero desbloquear nuevas funcionalidades al conseguir puntos, para conocer más a otra persona.
 - Descripción: Al aumentar los puntos se puede hacer click en el botón de los mismos para ver más datos del usuario en cuestión.
 - C. de aceptación:
 - Con 5 puntos, se desbloquea la estatura
 - Con 10 puntos, se desbloquea el peso
 - Con 20 puntos, se desbloquea una segunda foto
 - Tamaño: L
 - Prioridad: SH
 - Estado: Done
 - Inicio: Sprint 6
 - Final: Sprint 6

- Como usuario 1, quiero ganar puntos jugando con el usuario 2, para describir cómo de compatibles somos.
 - Descripción: Implementación de un método para almacenar puntos al jugar una partida.
 - C. de aceptación:
 - Usuario gana puntos al completar las partidas cooperativas.
 - Se registran los puntos obtenidos por la pareja
 - Los puntos de una pareja son visibles en la ventana del chat
 - Tamaño: M
 - Prioridad: SH
 - Estado: Done
 - Inicio: Sprint 5
 - Final: Sprint 5

- Como usuario precavido, quiero poder eliminar mi cuenta para no dejar rastro en la página web.
 - Descripción: Opción de eliminar cuenta en la visión del perfil que elimine los datos del Firebase.
 - C. de aceptación:
 - Se eliminan los datos de chats, usuario, juegos y todo lo relacionado con el usuario en Firebase.
 - Los usuarios que tenían chats con el perfil eliminado reciben un mensaje de que se ha eliminado
 - Tamaño: M
 - Prioridad: SH
 - Estado: Done
 - Inicio: Sprint 6
 - Final: Sprint 6

- Como usuario frecuente, quiero tener mis chats y la opción de jugar en la pantalla de inicio para que sea lo más fácil posible acceder a ellos.
 - Descripción: La pantalla de inicio muestra los chats en primer lugar, con la opción de invitar a jugar a dicho usuario. Los botones deben estar accesibles para el usuario

- C. de aceptación:
 - La pantalla de inicio muestra accesos directos a chats activos y a la opción de iniciar partida
 - El botón “jugar” estará visible y accesible sin tener que navegar por menús adicionales.
 - Todos los usuarios con los que se tienen chats activos serán accesibles desde la pantalla de chat
- Tamaño: XS
- Prioridad: SH
- Estado: Done

- Como usuario versátil, quiero que la página web funcione y se vea igual en todos los dispositivos.
 - Descripción: La página debe verse correctamente en todo tipo de dispositivos, sin importar las dimensiones de la pantalla:
 - C. de aceptación:
 - Todos los botones y funcionalidades son accesibles desde pantallas móviles.
 - Tamaño: M
 - Prioridad: CH
 - Estado: Done
 - Inicio: Sprint 6
 - Final: Sprint 6

- Como usuario, quiero salir de los juegos para dejar de jugar.
 - Descripción: Implementación de un método que dé la opción de abandonar el juego actual de forma sencilla y segura.
 - C. de aceptación:
 - Usuario puede salir del juego en cualquier momento desde la interfaz
 - Botón de salida claramente visible y accesible
 - Sistema guarda el estado del juego
 - Tamaño: S
 - Prioridad: CH
 - Estado: Done
 - Inicio: Sprint 4
 - Final: Sprint 4

- Como usuario, quiero que la aplicación tenga un nombre llamativo y original para que me guste.
 - Descripción: Definir nombre atractivo y único para la aplicación
 - C. de aceptación:
 - Fácil de recordar y visualmente llamativo
 - Refleja el espíritu de la aplicación
 - EL nombre es original y no coincide con otras aplicaciones de citas
 - Aprobado por todo el grupo
 - Tamaño: XS
 - Prioridad: CH
 - Estado: Done

- Inicio: Sprint 3
 - Final: Sprint 3
- Como usuario precavido, quiero salir de mi perfil sin necesidad de desinstalar la aplicación para más seguridad.
 - Descripción: Implementación de una opción para cerrar sesión o desvincular el perfil del dispositivo sin necesidad de desinstalar la aplicación, reforzando la seguridad y el control del usuario.
 - C. de aceptación:
 - Posibilidad de cerrar sesión en un solo paso.
 - Opción claramente visible y accesible desde la interfaz.
 - Tamaño: M
 - Prioridad: WH
 - Estado: Done
 - Inicio: Sprint 2
 - Final: Sprint 2
- Como usuario, quiero eliminar chats pasados para que dejen de estar ahí.
 - Descripción: Implementación de una opción de borrado, para mantener el chat limpio y controlado.
 - C. de aceptación:
 - Usuario puede eliminar chats desde el propio chat
 - Sistema elimina los chats sin afectar a otras partes del perfil
 - Se eliminan los datos del Firebase
 - Opción de eliminar es clara y accesible
 - Tamaño: S
 - Prioridad: WH
 - Estado: Done

(No implementadas: las que se han optado por abandonar)

- Como usuario precavido, quiero reportar a otros usuarios para evitar casos de acoso o violencia verbal.
 - Descripción: Implementación de un método con el que los usuarios pueden reportar diversas circunstancias, velando así por la seguridad y el respeto en la aplicación.
 - C. de aceptación:
 - Usuario puede acceder a la opción desde el perfil del usuario a reportar
 - Sistema registra el reporte
 - Los reportes se almacenan de forma segura y notifican a los moderadores
 - Tamaño: S
 - Prioridad: WH
 - Estado: Todo, no se ha implementado
- Como usuario precavido, quiero bloquear a otros usuarios para evitar interacciones no deseadas.
 - Descripción: Implementación de un método con el que los usuarios pueden bloquear/desbloquear a otros usuarios.

- C. de aceptación:
 - Usuario puede acceder a la opción desde el perfil del usuario a bloquear/desbloquear
 - Una vez bloqueado,
 - Tamaño: S
 - Prioridad: WH
 - Estado: Todo, no se ha implementado
- Como usuario 1, quiero poder enviar emoticonos y mensajes predeterminados al usuario 2 mientras estamos jugando.
 - Descripción: Implementación de un método que permite enviar emoticonos y mensajes predefinidos durante la partida, para facilitar la comunicación rápida y no perder la conexión durante el juego.
 - C. de aceptación:
 - Usuario puede seleccionar y enviar emoticonos / mensajes predeterminados desde la interfaz del juego
 - El sistema muestra correctamente los mensajes al otro jugador en tiempo real.
 - Tamaño: L
 - Prioridad: WH
 - Estado: Todo, no se ha implementado
 - Como usuario, quiero buscar entre mis chats para encontrar a la persona que busco.
 - Descripción: Implementación de un método de búsqueda de usuarios dentro de la ventana de chats.
 - C. de aceptación:
 - Usuario puede buscar por nombre
 - Resultados muestran las personas con ese nombre
 - Tamaño: S
 - Prioridad: CH
 - Estado: Todo, no se ha implementado

2.3. Sprint Reviews.

Aquí se muestran las sprint reviews que han sido realizadas a lo largo de los 6 sprints que han guiado el proyecto.

- **SPRINT 1**
 Identificado algunas historias de usuario que tenemos que modificar Tenemos un proyecto de GitHub bien organizado en distintas ventanas
 Tenemos que establecer un punto de partida del proyecto
- **SPRINT 2**
 En este sprint hemos desarrollado la arquitectura básica de la página web, así como un sistema de inicio de sesión local para poder crear y guardar usuarios. También se ha

empezado a implementar una forma de guardar datos adicionales de cada usuario como nombre, hábitos o foto.

- **SPRINT 3**

En este sprint hemos hecho una nueva página con el fin de sustituir a la anterior, y así poder tener un chat online y la posibilidad de almacenar los usuarios en línea, usando Firebase. Asimismo, hemos implementado un nuevo sistema mediante el cual se pide al usuario iniciar sesión o registrarse, para así poder ver la página con todos los botones. Hablando de los botones, estos han sido implementados de forma que abren diversas ventanas, como la Lista de Usuarios, el Chat, y el botón de perfil.

Centrándonos en la Lista de Usuarios, se trata de una ventana que muestra todos los id de los usuarios existentes, al lado de cada nombre hemos configurado un botón para copiar el id sin complicaciones. En la ventana de Chat, la cual ha sido el principal objetivo en este sprint, la hemos desarrollado mediante Firebase, donde almacenamos los usuarios, los chats y los mensajes. En ella hemos implementado un sistema mediante el cual un usuario puede añadirte y enviarte un mensaje, haciendo que aparezca el primero en la lista de mensajes. Por su parte, el botón de perfil permite abrir un mini menú con dos opciones, cerrar sesión y ver perfil; este último botón abre otra ventana flotante con dos menús, ver perfil para revisar todos tus datos, y editar perfil, donde el usuario podrá editar toda la información que ha puesto en su perfil. La información del perfil contiene un área de foto, una zona para escribir a mano intereses y demás, así mismo el usuario puede introducir su edad y rellenar una lista de opciones (deportista, lector, gamer...).

Finalmente, hemos buscado un nombre para la web, diseñado un logo y cambiado la paleta de colores.

- **SPRINT 4**

En este sprint hemos terminado de pulir detalles visuales de la página de inicio, como los colores o la distribución de los botones para moverte por las distintas funciones más fácilmente. Además, hemos añadido la opción de jugar a distintos juegos dentro de los chats, por ahora se tratan del conecta 4 y 2 verdades y 1 mentira, para conocer más a la otra persona y entretenerte. Para seguir, hemos implementado la opción de añadir imágenes como foto de perfil para los usuarios y que así se conozcan. A continuación, conseguimos crear un dominio web para el acceso de cualquier persona a la página web. Para pulir los últimos detalles, hemos corregido algunos errores generados en el anterior sprint que no habían sido descubiertos, además de los que hemos encontrado en las nuevas implementaciones.

Finalmente, hemos cambiado las historias de usuarios, tras votar un cambio en las ideas principales de la página web. Hemos decidido cancelar la necesidad de jugar un mínimo de partidas antes de poder hablar con un usuario, al ser el principal objetivo de la página conocer a otras personas, no jugar con ellas.

- **SPRINT 5**

Durante este sprint, hemos finalizado todas las tareas que nos hemos planteado. En cuanto a la sección de juegos, hemos implementado un sistema de puntos, que, tras finalizar una partida se suman los puntos correspondientes y se muestran los puntos totales en un banner dentro del chat. Además, hemos añadido un nuevo juego, el de dos verdades y una mentira. Pulsando el botón de juego que se encuentra dentro del chat, se crea la solicitud, en el que el usuario

escribe los enunciados y elige cual es la mentira. Será misión del otro usuario elegir cual proposición cree que es la mentira, una vez enviada la solución se sumarán los puntos correspondientes.

Por otra parte, hemos mejorado la estética de la interfaz, cambiando la estructura de la página principal y añadiendo widgets decorativos. Además de eso, hemos añadido easter eggs en toda la página web, para hacerla más apelativa y entretenida.

En cuanto al sistema del scroll, hemos mejorado el sistema de descubrir personas. Para ello hemos añadido para cada perfil un campo de género y gustos e intereses. Con esto, se filtra la búsqueda automáticamente según los intereses de cada uno, y solo muestra los usuarios que correspondan con tus gustos.

Finalmente, hemos aplicado las opiniones y las recomendaciones y opiniones que nos han comentado los otros grupos en sus RTF.

- **SPRINT 6**

Durante este sprint hemos logrado finalizar todas las tareas propuestas, terminando así con todas los requisitos que teníamos en nuestro MVP, además de muchas otras implementaciones adicionales. La página web ha mejorado exponencialmente en detalles que mejoran la comodidad y versatilidad de algunas funciones esenciales para el confort de los usuarios. Al principio del sprint decidimos hacer una planificación como veníamos haciendo últimamente, lo cual siempre ha sido una mejora en organización y orden en la realización de las tareas, y en esta ocasión, no ha sido distinto. Esta planificación fue ambiciosa, aún así, hemos sido capaces de realizar todo lo que nos propusimos con éxito.

En este sprint hemos añadido una mayor funcionalidad a los puntos para ver más información de los usuarios con los que interactúas, como puede ser altura, otra foto, etc.

Hemos añadido la opción de bloquear y desbloquear a otros usuarios en caso de no querer interactuar con ellos. Junto a esta opción, hemos añadido también la opción de poder borrar tu cuenta en caso de no querer seguir usando la web.

En el apartado de creación de cuentas e inicio de sesión, hemos implementado las contraseñas, para hacer más seguros los perfiles. Además, para poner a prueba la capacidad del servidor, hemos añadido 100 usuarios con un script, teniendo un resultado exitoso.

Para finalizar, en este sprint hemos hecho la presentación de la defensa final del proyecto, junto a la memoria del trabajo, la cual está casi finalizada a falta de algunos detalles.

En estas revisiones se refleja cómo al inicio del proyecto no conocíamos del todo la dinámica scrum ni la razón de hacer las cosas. Sin embargo, con el paso de los sprints se aprecia como las revisiones van siendo más concretas y extensas, mostrando una mejor imagen del incremento de ese sprint.

2.4. Sprint Retrospectives.

Aquí se muestran las sprint retrospectives que han sido realizadas a lo largo de los 6 sprints que han guiado el proyecto.

- **SPRINT 1**

- Hemos creado las historias de usuarios y posibles perfiles de usuarios de la app/web
- Hemos introducido todas las historias de usuario en el GitHub, clasificándolas por tamaño estimado (tiempo de implementación) y prioridad.

- **SPRINT 2**

Cosas a mejorar:

- Hemos subestimado el tiempo que teníamos y hemos asignado al sprint 2 más tareas de las que hemos podido hacer.
- Hemos usado poco GitHub y las facilidades que ofrece, sobre todo el kanban del sprint y el calendario.
- No había un objetivo claro del sprint.
- Hemos dividido las historias en tareas demasiado grandes, que requerían de varios días para ser completadas.
- Hay que centrarse más en la funcionalidad principal del proyecto.

Para mejorar estos aspectos, hemos decidido tomar las siguientes medidas:

- Coger menos historias de usuario en este sprint y medir la velocidad del equipo scrum en líneas de código y tamaño de las historias realizadas por horas de trabajo, para que nos sea más fácil estimar nuestra capacidad de incremento en el futuro.
- Apoyarnos más en la tabla kanban para organizarnos, así como utilizar distintas branches a la hora de trabajar en paralelo.
- Definir el objetivo del sprint desde el primer día del mismo.
- Dividir las historias de usuario en más tareas más pequeñas, para que sean más cómodas de realizar y no nos quedemos tan atascados. Si vemos que una tarea se está alargando demasiado, intentar dividirla en subtareas separadas en distintos pushes para que el resto del equipo pueda colaborar en las mismas.
- Elegir historias más dirigidas a la parte visible y llamativa del proyecto en vez de centrarnos en la funcionalidad.

Además, de los riesgos tratados en la gestión de riesgos, hemos detectado que se ha empezado a cumplir el número 6 al estimar mal el tamaño de las historias de usuario. De momento se tomarán las medidas explicadas más arriba, pero de persistir el problema en futuros sprints se revalorará el riesgo y la forma de actuar ante el mismo.

- **SPRINT 3**

A diferencia del sprint anterior, en el que habíamos subestimado el tamaño de las tareas y el tiempo que nos iban a tomar, hemos conseguido organizarnos de una manera más realista que nos ha permitido realizar todas las tareas de una manera satisfactoria. Tuvimos una reunión el Martes 28 de octubre de 2025 donde nos pusimos al día del trabajo que había hecho cada uno y pudimos cooperar de una manera bastante eficiente. No nos ha quedado ni una sola tarea por hacer, por tanto hemos aprovechado bien el tiempo y la organización realista con unas pocas historias de usuario que cumplir ha sido efectiva.

Cosas a mejorar:

- Falta de comunicación a la hora de repartirse las tareas.
- Ausencia de pruebas unitaria y control de calidad.
- Poca modularidad, archivos muy extensos.
- Pocos daily-meetings durante los sprints, que se traduce en falta de coordinación dentro del equipo.

Cómo mejorarlas:

- Para mayor comunicación, hemos decidido realizar más reuniones diarias para revisar lo que ha realizado cada uno hasta el momento. Además de una reunión semanal más extensa para realizar avances más complejos en grupo, y con la opinión y aportación de todos en las decisiones más importantes.
- Dedicarle parte del actual sprint a las pruebas unitarias y al control de calidad.
- Utilizar nuevos scripts para las nuevas funciones del proyecto.

Además, hemos identificado que nuestra gestión de riesgos necesitaba algunos cambios, para ser más fiel a la situación real. Hemos decidido intercambiar el riesgo 7 y el 14 dado que la frecuencia de commits no está siendo un problema como pensamos, y en cambio el examen de Cálculo Diferencial sí que nos quitó un tiempo bastante significativo influyendo en el trabajo durante el sprint. Por ello elaboraremos un análisis de este último más extenso en vista a futuros exámenes de otras asignaturas.

● **SPRINT 4**

Durante este sprint hemos logrado cumplir nuestros objetivos, sin embargo, y como teníamos previsto en la gestión de riesgos, los exámenes de otras asignaturas han dificultado, sobre todo, la posibilidad de trabajar todos juntos al mismo tiempo, o hacer los daily meetings que solemos hacer en la cafetería para hablar de qué ha hecho cada uno hasta ahora. Este inconveniente ha generado mucha falta de comunicación, que por suerte no ha causado muchos problemas, pero influía en la fluidez del trabajo.

Cosas a mejorar:

- Falta de comunicación a la hora de repartirse las tareas.
- Pocos daily-meetings durante los sprints, que se traduce en falta de coordinación dentro del equipo.
- Intervalos de poco trabajo durante fechas de exámenes.

Cómo mejorarlas:

- Dado que no podemos cambiar las fechas de exámenes a nuestro gusto, la única solución posible es realizar reuniones telemáticas durante las fechas más densas de trabajo. Esto podrá solucionar el problema principal que hemos encontrado durante este sprint. Además, de usar las horas libres entre clases para hacer reuniones muy breves, que a pesar de su corta duración ayudan a la hora de ponerse de acuerdo en ciertas decisiones y tareas.

● **SPRINT 5**

En este sprint hemos conseguido completar todas las tareas propuestas sin que nos sobrara demasiado tiempo, y hemos conseguido progresar significativamente en la página web. Crear

la planificación detallada al inicio del sprint nos ha ayudado a organizarnos mejor con las tareas para ver cuales podían hacerse en paralelo para progresar más. Además, se nota como la autoasignación de tareas va mejorando dentro del grupo, lo que se refleja en la productividad del grupo. Por último, el tamaño y duración estimado de las tareas cada vez es más adecuado, lo que ayuda a la hora de no dejar tareas a medias.

Cosas a mejorar:

- A la hora de hacer el merge con el dos verdades y una mentira, hubo errores en el GitHub que no supimos como tratar, que se tradujeron en marcas automáticas del GitHub en el código que no sabíamos que estaban ahí y nos costó identificar como problema.

Cómo mejorarlas:

- Si se repite, antes de hacer el merge automático con GitHub hacerlo a mano para saber en todo momento qué partes del código se han cambiado.

● SPRINT 6

En este último sprint hemos conseguido completar también todas las tareas que nos propusimos en un principio, por lo que podemos decir que en el estado actual hemos ganado bastante experiencia a la hora de estimar la cantidad de trabajo que podemos realizar y el tamaño de cada tarea. Otra cosa en la que hemos notado una mejora significativa es en la autoasignación de tareas, de forma que ya no hacemos trabajo duplicado en ningún momento. Además, hicimos bien al tener muy en cuenta a la hora de planificar el sprint la presentación del proyecto y la realización de la memoria que iban a consumir bastante tiempo en relación con otras tareas. Por último, hemos mejorado respecto al sprint anterior la forma de realizar los merges, consiguiendo hacerlos a mano y que GitHub no añadiera marcas que luego causarían errores.

Cosas a mejorar:

- Aunque hayamos mejorado la forma de hacer los merges, todavía no conocemos bien cómo funciona del todo, por lo que ha habido bastantes problemas de desinformación y no saber cómo hacer las cosas.

Cómo mejorarlas:

- Para futuros proyectos, investigar sobre los detalles del merge de GitHub y sus funcionalidades (al menos las principales) para juntar nuestro trabajo de forma mucho más ágil.

Estas retrospectivas reflejan los cambios que ha sufrido el equipo scrum a lo largo del desarrollo del proyecto. Así, se ve cómo las primeras retrospectivas son más vagas (la del sprint 1 ni siquiera es retrospectiva), pues todavía no entendíamos la forma de hacerlas correctamente ni el objetivo de las mismas. Sin embargo, según van pasando las semanas se aprecia cómo evolucionan hacia algo más completo y con una estructura definida. Para ello, hemos realizado reuniones en la cafetería al finalizar cada sprint, en las intentábamos identificar problemas que habíamos afrontado durante esas dos semanas así como la forma de evitarlos en el futuro.

2.5. Sprint Plannings.

Aquí se muestran las planificaciones que hemos documentado a lo largo del desarrollo del proyecto.

- **SPRINT 5**

En este sprint nos hemos centrado principalmente en 3 historias de usuario:

- Como usuario 1, quiero jugar a juegos cooperativos para poder conocer más al usuario 2: centrada en el desarrollo e introducción en la web del juego de dos verdades y una mentira. Al darle al botón de juegos ya existente en la ventana de chat se debe mostrar un desplegable con los dos juegos en vez de enviar la invitación del cuatro en raya. Después, de hacer click en uno de los dos juegos se envía una invitación de dicho juego, que debe mantener la funcionalidad actual. El juego debe funcionar de forma que cada usuario escriba dos verdades y una mentira sobre si mismo, eligiendo cual es la mentira. Cuando ambos usuarios hayan enviado sus tres opciones, el otro usuario debe tener la opción de elegir la mentira, y al hacerlo saber la solución correcta y volver a la pantalla de chat.

Con el mvp ya alcanzado esto pretende ampliar una de las dos partes principales de la página web, que son los juegos y la función de chatear. Queremos que haya varias opciones diferentes de juegos para que no se hagan repetitivos y siga siendo divertido jugar a los mismo durante un tiempo.

- Como usuario 1, quiero ganar puntos jugando con el usuario 2 para ver cómo de compatibles somos: se debe implementar un sistema que cada vez que se complete un juego con otro usuario sume puntos a ese chat para que cuanto más se juegue más puntos se tenga. Además, los puntos se deben mostrar en la ventana de chat para que los usuarios correspondientes puedan verlos.

Al igual que la historia anterior, esta pretende ampliar la parte de jugabilidad para que sea más llamativa. Además se enfoca más a la parte de encontrar a tu pareja ideal con un sistema para ver claramente con qué personas conectas mejor.

- Como usuario general, quiero definir mis intereses amorosos para poder encontrar una pareja adecuada para mi: se debe implementar un sistema tanto al crear como al modificar usuario para elegir los intereses amorosos entre hombre mujer y ambos. Además, en la lista de usuarios se deben mostrar solo perfiles que sean compatibles con tus gustos, para lo que se tiene que añadir la opción de si eres hombre o mujer. Esta es una parte esencial de la página web, para que cada usuario tenga una experiencia mucho más personalizada.

Además aunque no se trate de una historia de usuario existente, nos hemos propuesto refactorizar la lista de usuarios para que tenga un formato más llamativo. Para ello queremos cambiar el nombre a "Descubre gente", así como mostrar a los usuarios en cuadrados grandes en vez de cada uno en una línea, con la foto mucho más visible. También hemos planteado la idea de cambiar el botón de copiar nombre por uno de chatear directamente, para que sea más cómodo.

Por último, revisaremos los documentos del GitHub según lo comentado en las rtf's del sprint anterior. También arreglaremos algunos errores que detectamos pero no nos dio tiempo a tratar, y seguiremos desarrollando pruebas unitarias para nuestro código.

- **SPRINT 6**

En este sprint, nos vamos a centrar principalmente en 3 historias de usuario:

- Como usuario, quiero que mi perfil tenga una contraseña para poder poner la cuenta en distintos aparatos: pretendemos conseguir que al registrar un usuario te pida una contraseña, de esa manera, a dicho usuario se le queda una contraseña asociada, que necesitarás a la hora de iniciar sesión. Con esto queremos garantizar la privacidad y seguridad de cada usuario. En la mini ventana de registrar aparecerá un campo de contraseña, que estará codificado. Análogo en la de iniciar sesión.
- Como usuario, quiero desbloquear nuevas funcionalidades al conseguir puntos, para conocer más a otra persona: Vamos a añadir opciones que solo se desbloquean al alcanzar cierta cantidad de puntos. Por ejemplo, ciertas características de cada usuario, como el peso o la estatura. Para ello, antes de conseguir la cantidad de puntos necesarios, en la parte de ver perfil, no aparecerán dichas características. Una vez consigan los puntos, se actualizará. Por otra parte, a cada usuario se le pedirá como mínimo subir dos fotos a su perfil. De esos solo se mostrará una como foto de perfil, mientras que la otra únicamente se desbloqueará al alcanzar los puntos necesarios.
- Como usuario versátil, quiero que la página web funcione y se vea igual en todos los dispositivos: la interfaz de la página web se adaptará al dispositivo que esté usando el usuario, para evitar errores visuales y estéticos. Haremos que el tamaño de los widgets decorativos y la mini ventana de bienvenida se adapte al tamaño de cada dispositivo, para que se mantenga la estética de la página principal.

Además, queremos que el producto esté cerrado y funcional, por lo que nos centraremos en solucionar posibles errores y aspectos de la página web para conseguir un resultado final más pulido. Así, revisaremos qué pasa al cambiar los gustos, limitaremos peso y alturas para no poder introducir valores irracionales, y solucionaremos errores visuales como la foto de la ventana de chat al entrar por primera vez en la página web. También permitiremos iniciar sesión con el nombre de usuario en vez de con el email, e introduciremos 100 usuarios a Firebase para ver cómo se comporta la página web bajo un flujo de datos más grande.

Hasta que realizamos las RTFs en el sprint 4 no nos dimos cuenta de que debíamos realizar la planificación por escrito. Hasta entonces nos bastábamos del kanban del sprint como planificación, para mostrar qué historias habíamos elegido afrontar, así como en qué subtareas las dividíamos. Con todo esto, sí que notamos una mejora en la forma de organizar los tableros kanban según íbamos ganando experiencia con el paso de los sprints. Esto se notó sobre todo al empezar a usar medidas de tamaño y duración también para las subtareas, con lo que pudimos disminuir las veces que se quedaban tareas a medias evitando solaparnos en las mismas y retrasos en el progreso del proyecto. Además, realizar las retrospectivas nos ayudó a ser progresivamente más realistas con el número de historias de usuarios escogidas al prever qué cosas podían obstaculizar el desarrollo del proyecto. Así, cada vez nos quedaban menos tareas a medias al finalizar los sprints, hasta que en el quinto conseguimos acabarlas todas. También se notó una evolución notable entre el sprint 2 y el 3, en el

aspecto de qué historias de usuario escoger. Al realizar la retrospectiva nos dimos cuenta de que nos habíamos centrado en aspectos de la página web necesarios pero muy poco visibles, por lo que el producto no reflejaba el trabajo realizado. Así, en sprints posteriores decidimos enfocarnos en aspectos más llamativos, como el chat, los juegos o la funcionalidad de descubrir gente. En cuanto a las planificaciones escritas como tal, no notamos que nos ayudaran demasiado en el desarrollo del sprint, pues ya nos habíamos hecho a la dinámica de trabajar sobre el kanban.

2.6. Product Backlog.

Es importante poder trazar cual ha sido la evolución del product backlog a lo largo del proyecto, de manera que se pueda ver en qué momento entraron y/o salieron las historias de usuario del backlog, qué modificaciones han sufrido y qué motivo cada una de estas acciones.

Las modificaciones de estas historias de usuario durante el desarrollo del proyecto son debido a nuestro desconocimiento inicial de la estructura y finalidad de las historias de usuario. Por tanto, hicimos pequeñas modificaciones para conseguir solidez en las historias.

Historias de usuario modificadas durante el proyecto:

- Como usuario 1, quiero chatear con otros usuarios para poder conocernos mejor.
 - Su antiguo nombre era: “Como usuario 1, quiero chatear con otros usuarios una vez hayamos avanzado en el juego para poder conocernos mejor.”
 - Tiempo: Sprint 3
- Como usuario 1, quiero que mi perfil tenga una contraseña para poder poner la cuenta en distintos aparatos
 - Fue dada como Done durante el Sprint 2, pero decidimos eliminar el código para que las pruebas unitarias fuesen de una forma más fluida (Crear usuarios sin necesidad de dar una contraseña). Posteriormente, en el Sprint 6 para conseguir el Terminado de nuestro proyecto, lo añadimos.
 - Tiempo: Sprint 2 y Sprint 6
- Como usuario 1, quiero ganar puntos jugando con el usuario 2 para descubrir como de compatibles somos
 - Su antiguo nombre era: “Como usuario 1, quiero ganar puntos jugando con el usuario 2, para avanzar.”
 - Modificado: Sprint 5

Historias de usuario eliminadas:

- Como usuario 1, quiero enviar una solicitud al usuario 2 para cambiar de juego
 - Descripción: Implementación de un método de solicitud para cambiar el juego durante una sesión.
 - C. de aceptación:
 - Usuario 1 puede emitir una solicitud de cambio dentro de la partida

- Usuario 2 recibe una notificación con opción de aceptar o rechazar
 - El sistema registra la solicitud y su estado
- Como usuario 1, quiero aceptar o rechazar la solicitud del usuario 2 de cambiar de juego para seguir jugando.
 - Descripción: Implementación de un método que permita al usuario 1 responder la solicitud del usuario 2 para cambiar de juego durante una sesión activa
 - C. de aceptación:
 - Usuario 1 puede aceptar o rechazar la solicitud de cambio de juego
 - Solicitud se muestra en una notificación o pantalla emergente
 - Sistema registra la decisión y actúa en consecuencia
- Como usuario 1, quiero avanzar lo suficiente en el juego con el usuario 2 para poder hablar y quedar con él.
 - Descripción: Implementación de una mecánica que vincule el progreso de dos usuarios. Una vez hayan conseguido suficientes puntos superando los distintos niveles del juego, se debe abrir la opción de chatear entre ellos
 - C. de aceptación:
 - Interfaz que respete la estética del juego.
 - Sistema de puntos según se vayan superando niveles.
 - Una vez llegado a un mínimo prefijado de puntos, opción para empezar a chatear.
- Como usuario 1, quiero jugar en línea con el usuario 2 para ver lo compatibles que somos.
 - Era igual a otra historia de usuario: Como usuario 1, quiero jugar a juegos cooperativos para conocer más al usuario 2.
 - Las hemos fusionado
- Como usuario, quiero buscar perfiles según distintos filtros para no perder tiempo.
 - Era igual a otra historia de usuario: Como usuario general, quiero poder definir mis intereses amorosos y buscar perfiles según dichos intereses para poder lograr una pareja adecuada para mí.
 - Las hemos fusionado

2.7. Sprint Backlog.

En esta sección se tratará la organización de cada uno de los distintos sprint backlogs, separando las historias de usuario elegidas con sus subtarefas, otras cuestiones que no dependían de ninguna historia de usuario, y tareas que quedaron sin hacer en sprints anteriores. Se escribirán después de las historias de usuario y subtaréas de tenerlo su tamaño y prioridad en el formato (TAMAÑO/PRIORIDAD)

● SPRINT 1

Este sprint fue un inicio para organizarnos y crear las primeras historias de usuario, por lo que no creamos ningún sprint backlog.

- **SPRINT 2**

En este sprint tratamos las siguientes historias divididas en sus correspondientes subtareas:

1. Como usuario, quiero que mi perfil tenga una contraseña para poder poner la cuenta en distintos aparatos. (S/MH)
 - Impedir que se puedan crear dos usuarios con el mismo nombre.
 - Botón en la pantalla de inicio que abra una ventana con las opciones de registrarse e iniciar sesión.
 - Opción de iniciar sesión.
 - Opción de registrarse.
 - Crear un documento al que pueda acceder la página web para guardar usuarios y contraseñas.
2. Como usuario nuevo, quiero crear un perfil para controlar todos mis datos tales como foto, nombre, hábitos o intereses.
 - Una vez se haya dado al botón de crear en la pantalla de registro, crear una encuesta que pregunte el nombre
 - Al registrarse, guardar los datos de cada usuario en un documento para que la página web pueda acceder
3. Como usuario, quiero ver perfiles de otros usuarios.
 - Cuando el usuario haga scroll hacia abajo, se muestran nuevos nombres de usuarios diferentes
 - Ver la foto y otros datos del usuario en cuestión
 - Arreglar la lista de usuarios
 - Implementar un sistema de scroll infinito hacia abajo en la página web
 - Tener un botón en la pantalla de inicio que cuando se apriete muestre la lista de los usuarios guardados en el servidor
4. Como usuario nuevo, quiero crear un perfil para controlar todos mis datos tales como foto, nombre, hábitos o intereses. (L/MH)
 - Cambiar preferencias (arreglar la parte de la foto)
 - Añadir a la encuesta las opciones de foto, hábitos, intereses románticos y personalidad

Sumadas a otras tareas adicionales

1. Explicar el "Terminado" de nuestro proyecto
2. Cambiar historias de usuario 1 y 10
3. Crear documento de gestión de riesgos
4. Crear wiki
5. Definir MVP, el producto mínimo viable
6. Describir en el README la información del proyecto relevante a los usuarios
7. Crear la arquitectura de una página web sobre la que poder desarrollar la aplicación

Así en este sprint nos centramos en la parte del perfil y el inicio de sesión y registro de nuevos usuarios en local. Además, dimos mucha importancia a la documentación del proyecto, creando todos los archivos necesarios al inicio del mismo.

- **SPRINT 3**

En este sprint tratamos las siguientes historias divididas en sus correspondientes subtareas:

1. Como usuario 1, quiero chatear con otros usuarios para poder conocernos mejor.
 - Interfaz de que te ha llegado un mensaje
 - Campo para escribir y opción para mandar los mensajes escritos
 - Tener un enlace que abra una ventana de chat
 - Poder chatear con usuarios específicos
2. Como usuario, quiero ver perfiles de otros usuarios.
 - Tener un botón en la pantalla de inicio que cuando se apriete muestre la lista de los usuarios guardados en el servidor.
 - Arreglar la lista de usuarios
 - Arreglar el guardar del userID
 - Cuando el usuario haga scroll hacia abajo, se muestran nuevos nombres de usuarios diferentes
 - Implementar un sistema de scroll infinito hacia abajo en la página web
 - Ver la foto y otros datos del usuario en cuestión
3. Como usuario, quiero que la web tenga servicios online para poder acceder desde cualquier ubicación y comunicarme con otros usuarios.
 - Investigar el método a utilizar

Sumadas a otras tareas adicionales:

1. Mover la funcionalidad de los botones flotantes de Chat y Lista de Usuarios a los botones de la barra de navegación
2. Empezar a hacer alguna prueba unitaria sobre las funcionalidades más importantes
3. Crear un esquema de la interfaz

Y algunas que nos quedaron de sprints anteriores:

1. Añadir a la encuesta las opciones de foto, hábitos, intereses románticos y personalidad.
2. Elegir nombre.
3. Formatear el GitHub

- **SPRINT 4**

En este sprint tratamos las siguientes historias divididas en sus correspondientes subtareas:

1. Como usuario 1, quiero jugar a juegos cooperativos para conocer más al usuario 2. (XL/MH)
 - Subir el juego de dos verdades y una mentira para que se pueda jugar en línea a través del chat (L/CH)
 - Implementar en local un juego de dos verdades y una mentira (M/SH)
 - Crear el juego del cuatro en raya (XL/SH)
 - Interfaz con piezas de distintos colores y tablero (S/CH)
 - Permitir jugar a personas en el servidor (L/CH)
 - Implementar la lógica, que se pueda jugar en local (L/SH)
2. Como usuario, quiero ver perfiles de otros usuarios. (L/MH)
 - Ver la foto y otros datos del usuario en cuestión (M/SH)
 - Implementar un sistema de scroll infinito hacia abajo en la página web (S/CH)

- Cuando el usuario haga scroll hacia abajo, se muestran nuevos nombres de usuarios diferentes (S/CH)
- 3. Como usuario nuevo, quiero crear un perfil para controlar todos mis datos tales como foto, nombre, hábitos o intereses.
 - Al registrarse, guardar los datos de cada usuario en un documento para que la página web pueda acceder (M/SH)
- 4. Como usuario, quiero invitar a la gente para jugar juntos. (M/MH)
 - Integrar los juegos en la página web (XL/CH)
 - Botón en la ventana de chat para invitar a jugar (S/CH)
- 5. Como usuario, quiero salir de los juegos para dejar de jugar. (S/CH)

Sumadas a otras tareas adicionales:

1. RTF Grupo Feudalia (S/MH)
2. RTF Grupo Scrabble (S/MH)
3. RTF Grupo Logbait (S/MH)
4. ERROR cambiar nombre de usuario (L/CH)
5. Pruebas unitarias (XL/SH)
6. Crear un dominio para la página web (S/MH)

Y algunas que nos quedaron de sprints anteriores:

1. Cambiar preferencias (arreglar la parte de la foto) (XL/MH)
2. Añadir a la encuesta las opciones de foto, hábitos, intereses románticos y personalidad (L/MH)

● SPRINT 5

En este sprint tratamos las siguientes historias divididas en sus correspondientes subtareas:

1. Como usuario general, quiero poder definir mis intereses amorosos para poder lograr una pareja adecuada para mí (L/SH)
 - Mostrar en la lista de usuarios solo aquellos que se identifiquen con tu preferencia (L/SH)
 - Añadir la opción de intereses amorosos al perfil (Hombre, mujer, ambos) (M/MH)
 - Añadir opción de género al perfil (M/CH)
2. Como usuario 1, quiero ganar puntos jugando con el usuario 2, para descubrir como de compatibles somos (M/SH)
 - Interfaz en la ventana de chat que muestre los puntos de esa pareja (S/MH)
 - Sistema que añada puntos cada vez que se completa una partida de algún juego con otro usuario (M/MH)
3. Como usuario 1, quiero jugar a juegos cooperativos para conocer más al usuario 2. (XL/MH)
 - Implementar en local un juego de dos verdades y una mentira (M/SH)
 - Subir el juego de dos verdades y una mentira para que se pueda jugar en línea a través del chat (L/CH)

Sumadas a otras tareas adicionales:

1. Pruebas unitarias (L/SH)
2. ERROR Números al lado de los usuarios en el chat (L/CH)
3. Refactorizar lista de usuarios (M/SH)

- ERROR Al salir de la ventana de ver los detalles de un usuario se cierra a su vez la lista de usuarios (M/SH)
 - Cambiar la opción de copiar usuario por una que te permita directamente abrir el chat con dicho usuario (M/MH)
 - Cambiar el nombre de lista de usuarios a descubre gente (XS/MH)
 - No mostrar los usuarios con los que ya se tiene chats abiertos (S/CH)
 - Ver los usuarios como cuadrados con la foto en grande en vez de en una lista (L/SH)
4. Revisar lo comentado en rtf (M/MH)
- Historias de usuario (Grupo Feudalia) (M/MH)
 - Planificación (Grupo logbait) (M/MH)
 - Riesgos (Grupo Scrable) (M/MH)

● SPRINT 6

En este sprint tratamos las siguientes historias divididas en sus correspondientes subtareas:

1. Como usuario versátil, quiero que la página web funcione y se vea igual en todos los dispositivos (M/CH)
2. Como usuario, quiero que mi perfil tenga una contraseña para poder poner la cuenta en distintos aparatos. (L/MH)
 - Opción de registrarse. (S/MH)
 - Opción de iniciar sesión (S/MH)
 - Impedir que se puedan crear dos usuarios con el mismo nombre (S/MH)
 - Botón en la pantalla de inicio que abra una ventana con las opciones de registrarse e iniciar sesión (S/MH)
3. Como usuario precavido, quiero poder eliminar mi cuenta para no dejar rastro en la web (M/SH)
 - Eliminar los datos de Firebase (S/MH)
 - Comprobar que a los demás usuarios no les aparece el usuario eliminado (S/SH)
 - Mensaje de haber borrado la cuenta en los chats del usuario (M/WH)
4. Como usuario, quiero desbloquear nuevas funcionalidades al conseguir puntos, para conocer más a otra persona (L/SH)
 - Al aumentar los puntos, se pueden ver al pulsar sobre los mismos la altura y peso de los usuarios (M/SH)
 - Al aumentar puntos, poder ver más fotos de ese usuario (M/SH)
 - Introducir la opción de guardar dos imágenes en la página web (M/SH)
 - Introducir opción de estatura y peso al crear usuario (XS/CH)

Sumadas a otras tareas adicionales:

1. Poder iniciar sesión con el nombre de usuario y la contraseña (M/SH)
2. ERROR: Revisar que pasa cuando cambias los gustos (S/SH)
3. ERROR Arreglar foto aleatoria que sale en la ventana chat cuando no hay abierto ninguno (M/MH)
4. Limitar pesos y alturas falsas (XS/SH)
5. Añadir 100 usuario a la página web (M/MH)
 - Hacer un documento de texto con 100 usuarios con un formato específico (L/MH)
 - Añadir todos los usuarios al Firebase (M/MH)

6. Hacer obligatorio la introducción de la foto de perfil (XS/MH)
7. Implementar sistema de bloqueo / desbloqueo de usuarios

Y algunas que nos quedaron de sprints anteriores:

1. Explicar el "Terminado" de nuestros proyecto (M/MH)

2.8. Descripción del contenido del repositorio en GitHub.

Descripción del uso que se ha hecho de las características de GitHub para el desarrollo del proyecto: issues, projects, wiki, etc.

- **ISSUES**

Hemos utilizado el apartado de Issues para almacenar tanto las Historias de Usuario como las subtarefas de cada una. Nos sirve para ver qué tareas siguen abiertas y cuales ya han sido acabadas.

Su funcionalidad es básicamente informativa.

- **PROJECTS**

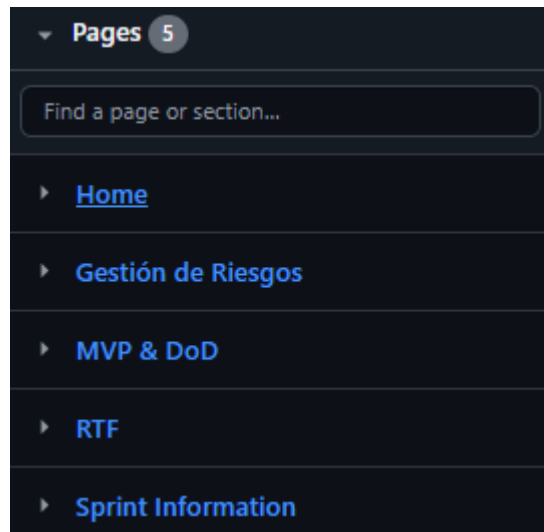
El apartado de Projects nos ha servido para crear el backlog con solo las Historias de Usuario. Además, nos ha servido para gestionar las tareas que queríamos realizar en cada Sprint. Tenemos un proyecto específico para cada Sprint (1-6) donde se separan las tareas según su estado: Todo, In Progress, To Test, Done. Cada tarea puede asignarse a uno o varios miembros del equipo, agilizando la organización del Sprint. Todos nuestros proyectos son estos:



Proyectos creados al final del sprint

- **WIKI**

La wiki nos sirve para presentar en GitHub nuestro proyecto tanto a nuestros compañeros como al profesor. Está dividido en 5 apartados:



Índice de la wiki

- **Home**

Nos permite mostrar los objetivos y características del proyecto de una manera divulgativa para captar la atención tanto de los otros miembros de IS como clientes potenciales.

- **Gestión de Riesgos**

Aquí nos encargamos de analizar y gestionar, como su propio nombre indica, los riesgos potenciales del proyecto, con dos características cada una, la probabilidad del riesgo y su impacto en nuestro trabajo.

Probabilidad: Frecuente / Probable / Ocasional / Remoto / Improbable

Gravedad: Despreciable / Menor / Serio / Crítico / Catastrófico

- **MVP y DoD**

Se centra en las condiciones que debe cumplir nuestro proyecto para ser considerado Terminado (DoD: Definition of Done) y MVP (Minimum Viable Product), es decir, lo mínimo que ha de presentar el proyecto para que cumpla la funcionalidad que se pretendía desde un inicio.

- **RTF**

Estas revisiones han sido realizadas en proyectos de compañeros de la asignatura.

Al grupo LogBait se le ha corregido la Planificación, revisión y retrospectiva de los Sprint.

Al grupo Scrabble se le ha corregido la Gestión de Riesgos.

Al grupo Feudalia se le ha corregido la solidez y estructura de las Historias de Usuario.

- **Sprint Information**

Al final de cada Sprint, se debía hacer una Sprint Review(tareas realizadas durante el sprint) y una Sprint Retrospective (¿que se ha de mejorar para el siguiente sprint?), esto está plasmado en este apartado.

- **TORTOISEGIT**

Esta aplicación no es en sí de GitHub, pero nos ha ayudado bastante para la creación y modificación de código. Mediante el “Tortoise” hemos sido capaces de probar localmente cambios del proyecto sin así interferir en el proyecto en sí. Una vez veíamos que funcionaba como se esperaba, se hacía push para incluirlo en el GitHub.

3. Otras tareas de gestión del proyecto.

[En este apartado se incluirá toda la información relativa a las tareas de gestión del proyecto que no están incluidas directamente en scrum, como han sido la gestión de riesgos o la realización de las revisiones técnicas formales. En el caso de la gestión de riesgos, deberá describirse no sólo el plan de gestión de riesgos, sino la manera en que se ha hecho el seguimiento de los mismos. En cuanto a las revisiones técnicas formales, deberá contarse, por un lado, de qué manera se han preparado las RTFs realizadas a los otros grupos, y por otro, las acciones correctivas llevadas a cabo con las RTFs recibidas.]

3.1. Gestión de riesgos

Este apartado de gestión de riesgos tiene como objetivo identificar, analizar y mitigar los posibles eventos que puedan afectar al desarrollo de nuestro proyecto. Esta sección describe los principales riesgos detectados, su probabilidad y gravedad, así como las medidas de reducción, supervisión y gestión propuestas, con el fin de aumentar la credibilidad y profesionalidad de nuestro producto.

Hemos creado una lista de riesgos, y los hemos categorizado según la probabilidad de que suceda:

- Falta de conocimientos extensos en cualquiera de los lenguajes utilizados
- No poder implementar la parte de juegos cooperativos en línea
- Falta de tiempo para poder desarrollar el software necesario en el cuatrimestre
- Incapacidad para poder comunicar dos usuarios entre ellos a través de un servidor
- Que se repita otro apagón durante una sesión crítica de programación
- Mala estimación del tiempo necesario o el tamaño de desarrollo de una historia de usuario
- Que todo el equipo tenga menos tiempo por motivo de exámenes o entregas
- Que la visión del proyecto sufra muchos cambios
- Que nuestro scrum master (el profesor) sufra una baja laboral
- Que un miembro del equipo deje la carrera
- Planificar sprints con tamaños de tareas poco realistas
- Desarrollo de funcionalidades incorrecto
- Ausencia de uno o varios miembros del equipo por enfermedad
- Poca frecuencia de commits, que se traduce en una gran dificultad a la hora de hacer merges
- Alta dependencia de IAs generativas
- Errores de compatibilidad entre el código de cada miembro del equipo
- Requisitos (o criterios de aceptación) mal definidos
- Que alguno de los integrantes del proyecto no pueda acceder a entornos de programación como Visual Studio

Una vez tenemos la lista de riesgos, utilizando la regla de Pareto, hemos desarrollado un plan de reducción, supervisión y gestión para los riesgos más significativos, con el fin de poder evitar y reaccionar ante la aparición de alguno de ellos de una forma mucho más eficiente.

Gravedad \ Probabilidad	Frecuente	Probable	Ocasional	Remoto	Improbable
Catastrófico	1	2	3	4	5
Crítico	6	7	8	9	10
Serio	11	12	13	14	
Menor	15		16		
Despreciable			17		18

Tabla de riesgos del proyecto clasificados por gravedad y probabilidad

- **Riesgo 1:** Falta de conocimientos extensos en cualquiera de los lenguajes utilizados
 - **Reducción:** Fomentar el aprendizaje previo mediante tutoriales, documentación oficial y pequeñas prácticas individuales antes de abordar tareas críticas del proyecto.
 - **Supervisión:** Revisar periódicamente el progreso de cada miembro para detectar dificultades técnicas y reasignar tareas si es necesario.
 - **Gestión:** Si después de 1 semana del último sprint, un miembro no puede alcanzar el nivel requerido, se redistribuirán sus tareas críticas a otros miembros y se le asignarán tareas de menor complejidad técnica, tales como documentación, testing, etc.
- **Riesgo 2:** No poder implementar la parte de juegos cooperativos en línea
 - **Reducción:** Diseñar una arquitectura modular que permita aislar la funcionalidad de red, facilitando pruebas tempranas y el uso de bibliotecas consolidadas.
 - **Supervisión:** Establecer hitos específicos para la comunicación cliente-servidor y realizar revisiones semanales del estado de la conexión y sincronización.
 - **Gestión:** Si para el sprint 4 el sistema cooperativo no se logra implementar a tiempo, se activará el plan alternativo: ofrecer un modo local, offline, por turnos en un mismo dispositivo, manteniendo la jugabilidad base pero eliminando la conectividad online.
- **Riesgo 3:** Falta de tiempo para poder desarrollar el software necesario en el cuatrimestre
 - **Reducción:** Elaborar una planificación realista desde el inicio, priorizando las funcionalidades esenciales y definiendo un calendario con márgenes para imprevistos. Dividir el trabajo en tareas pequeñas y alcanzables.
 - **Supervisión:** Revisar semanalmente el progreso del proyecto durante las reuniones de equipo y comparar el avance real con la planificación inicial.
 - **Gestión:** Si se detecta retraso significativo (superior al 30% en cualquier sprint), se eliminarán 3 funcionalidades secundarias (2 Could y 1 Should), seleccionadas por votación del equipo basándose en el valor que aporta para el producto, todas las Won't y se eliminarán características que no comprometan la funcionalidad principal, priorizando funciones clasificadas como Must. Además, el equipo dedicará 3 horas extra cada semana hasta recuperar el cronograma.

- **Riesgo 6:** Mala estimación del tiempo necesario o el tamaño de desarrollo de una historia de usuario
 - **Reducción:** Realizar una estimación colaborativa de tiempos y basarse en la velocidad observada de sprints anteriores usando nuestra sección de "Calendario".
 - **Supervisión:** Evaluar al final de cada sprint si las estimaciones fueron realistas y ajustar la metodología en función de los resultados obtenidos.
 - **Gestión:** Si en dos sprints consecutivos se detecta una desviación considerable (<70% completitud), se reestructurará el backlog priorizando funcionalidades "Must", se aplazarán las tareas Should y Could menos prioritarias y se eliminarán las Won't
- **Riesgo 7:** Que todo el equipo tenga menos tiempo por motivo de exámenes o entregas
 - **Reducción:** Planificar los sprints teniendo en cuenta los distintos exámenes y otros factores que pueden tener mucha influencia en el tiempo que todos los miembros del grupo invierten en el proyecto.
 - **Supervisión:** Revisar que la mayoría de las tareas propuestas dentro de un sprint se hayan podido terminar.
 - **Gestión:** Durante periodos de exámenes, la carga de trabajo se reducirá al 50%, enfocándose únicamente en resolver deuda técnica y bugs críticos, sin desarrollar ninguna nueva funcionalidad que no sea Must. Si el retraso que provoca este riesgo es significativo, se implementará la gestión del riesgo 3.3

Asimismo, hemos creado una planificación temporal que nos permite controlar y evitar los riesgos seleccionados, al inicio de los sprints revisamos el estado de dichos riesgos para poder actuar en caso de que alguno estuviera sucediendo. Esto se hace a la vez que la retrospectiva y la planificación del siguiente sprint, cada dos viernes después de clase.

3.2. RTF (Revisiones Técnicas Formales)

En cuanto a las RTF, nos hemos dividido en subgrupos dentro del equipo de trabajo para que cada subgrupo se dedique a hacer las RTF correspondientes de cada grupo distinto, esto es, subgrupo 1 se dedica a hacer las RTF del grupo LogBait, subgrupo 2 se dedica a hacer las RTF del grupo Scrabble, etc. Además de nuestras RTF, se adjunta las RTF realizadas por los demás grupos

Grupo LogBait

Revisor: Veneris

Revisado: Gestión de riesgos grupo Logbait

- No se aplica la regla de Pareto y se desarrollan medidas para todos los riesgos en vez de solo para los más graves y probables.
- No se establece un plan de supervisión para ninguno de los riesgos identificados
- Los planes de mitigación se pueden incluir como estrategias de prevención
- No se define una planificación temporal para revisar la gestión de riesgos durante los sprints
- La gravedad y probabilidad de los riesgos debería revisarse en base a la experiencia adquirida en los distintos sprints

Revisor: LogBait

Revisado: Planificación, revisión y retrospectiva Veneris

- Solo hay planificación en el backlog no en la Wiki
- Falta sprint goal en el sprint 2
- Algunas historias son demasiado grandes

Grupo Scrabble

Revisor: Veneris

Revisado: Planificación, retrospectiva y revisión de Scrabble

- Falta sprint goal de este sprint
- No se especifica ningún tipo de revisión de la gestión de riesgos. Si los riesgos especificados siguen estando correctos se debería ver reflejado en la retrospectiva, igual que si hace falta cambiar algo.
- En la retrospectiva entrar más a fondo en la parte de que se ha hecho mal

Revisor: Scrabble

Revisado: Gestión de riesgos

- Algunas medidas de gestión son genéricas y podrían detallarse más
- Es necesario concretar mejor los planes de contingencia, especificando qué se aplicaría si el riesgo finalmente ocurre

Grupo Feudalia

Revisor: Veneris

Revisado: Historias de usuario grupo Feudalia

En general, la mayoría de las historias de usuario están correctamente enfocadas en funcionalidades del sistema y cuentan con condiciones de aceptación adecuadas. Además, se destaca la diferenciación de tipos de usuario (casual, competitivo, estratégico), lo que permite comprender de manera más realista las necesidades específicas de cada perfil.

Cosas a mejorar:

- Hay historias que solo imitan la estructura, pero no representan un requerimiento funcional, como aquellas relacionadas con tareas del equipo, decisiones internas o conceptos abstractos.
- La organización de las historias de usuario en la wiki resulta confusa, ya que al inicio se destacan las historias completadas en el primer sprint. La página de historias de usuario, que debería consistir únicamente en la descripción de estas, no es el lugar adecuado para ello, y su inclusión provoca que algunas historias se repitan.
- Varias historias carecen de independencia, pues dependen de otras funcionalidades.
- Hay historias que carecen de claridad, aunque se pueden entender con las condiciones de aceptación, cada historia debería ser clara por sí misma.
- Algunas historias son demasiado largas o abarcan múltiples funcionalidades, lo que dificulta entender claramente su objetivo y puede complicar su desarrollo independiente.

Recomendaciones:

- Eliminar historias relacionadas con tareas del equipo, decisiones internas o conceptos abstractos, o colocarlas en otro apartado separados de las historias de usuario.
- Evitar lenguaje ambiguo o conceptos genéricos; ser concreto en lo que se quiere lograr.
- Si una historia incluye varias acciones o objetivos, dividirla en varias historias más pequeñas.

Hemos analizado cada historia de usuario, según las características:

- Independencia entre HU: ¿puede desarrollarse sin depender de otra?
- Claridad: ¿se entiende fácilmente?
- Relevancia para los clientes : ¿aporta valor para el usuario final?
- Verificabilidad: ¿puede comprobarse objetivamente una vez acabada?

Para ello, hemos evaluado cada historia de usuario en una tabla:

- E: Erróneo
- I: Incompleto
- P: Perfecto

Sprint 1 finalizadas (Prototipo inicial básico)

Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como jugador experimentado, quiero un control preciso del personaje para explorar el mapa, alcanzar recursos y evaluar zonas de combate	P	I	P	I
Como jugadora con experiencia en estrategia, quiero que mi personaje se ubique correctamente en la pantalla desde el primer momento	P	P	P	P
Como jugadora casual, quiero poder entrar al juego mediante un menú intuitivo y simple para poder jugar una partida	P	I	I	I
Como jugador, quiero recibir un estímulo visual al moverme por el entorno	I	I	I	E

Sprint 1

Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como desarrollador, quiero que quede organizado un repositorio de trabajo compartido para poder colaborar dentro del equipo (NO ES HISTORIA DE USUARIO)	P	P	E	P
Como equipo, queremos establecer los objetivos a cumplir en el siguiente sprint (NO ES HISTORIA DE USUARIO)	I	E	E	I
Como futuro jugador, quiero una descripción general del juego para saber de qué tratará	P	P	P	P
Como desarrollador del proyecto, quiero tener una idea clara del propósito y alcance de las mecánicas (NO ES HISTORIA DE USUARIO)	P	I	E	E
Como diseñador, quiero contar con un boceto visual del juego (NO ES HISTORIA DE USUARIO)	P	I	E	P
Como miembro del equipo, quiero que se establezca cómo se va a desarrollar el juego (NO ES HISTORIA DE USUARIO)	P	I	E	P

Usuario Persona 2 (Usuario Casual)				
Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como jugadora casual, quiero ver la cantidad de cada recurso para saber qué funcionalidades puedo usar	P	P	P	P
Como jugadora casual, quiero poder entrar al juego mediante un sistema de login	P	P	P	P
Como jugadora casual, quiero poder declarar combate a mis enemigos	I	P	P	P
Como jugador, quiero poder acceder al juego mediante un menú inicial que me permita jugar	P	P	P	P
Como jugador casual, quiero obtener el territorio del enemigo derrotado	I	I	I	P
Como jugador casual, quiero poder ganar una partida al conquistar territorios enemigos	I	E	P	I
Como jugador, quiero poder acceder a un menú de construcción	P	I	I	P
Como jugador, quiero enviar una orden para expandir mi territorio	E	I	I	P
Como jugador, quiero enviar mensajes a otros jugadores para planear alianzas o trueques	P	P	I	P

Usuario Persona 1 (Usuario Intensivo)				
Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como jugador competitivo, quiero ver en todo momento mis recursos para optimizar mi economía y planificar rutas estratégicas	I	P	P	P
Como jugador experimentado, quiero un control preciso del personaje para explorar, alcanzar recursos y evaluar zonas de combate (REPETIDA)	P	I	P	I
Como jugador competitivo, quiero mejorar y hacer más eficientes mis acciones para fortalecer ejército, edificios y recolección	E	I	P	E
Como jugadora con experiencia en estrategia, quiero que mi personaje se ubique correctamente al inicio (REPETIDA)	P	E	P	P
Como jugadora estratégica, quiero que el mapa refleje claramente áreas de combate y recursos	P	I	P	P
Como jugadora intensiva, quiero recibir feedback visual inmediato al recolectar recursos o interactuar con el mapa	I	I	P	P
Como jugador, quiero asignar aldeanos a tareas de recolección para automatizar la obtención de recursos	I	P	I	P

User Persona 3 (Usuario Tangencial)				
Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como jugadora interesada en expansión de imperios, quiero ver claramente cómo crece mi imperio (NO ES HISTORIA DE USUARIO)	E	E	I	E
Como jugadora que valora facilidad de uso, quiero una interfaz sencilla e intuitiva (NO ES HISTORIA DE USUARIO)	I	E	P	I
Como jugador, quiero poder construir edificios como leñadores o mineros	P	E	E	I
Como jugador, quiero que el progreso se transmita mediante mecánicas (NO ES HISTORIA DE USUARIO)	E	E	E	E
Como jugador, quiero poder construir una casa para aumentar mi población	P	I	P	P
Como jugadora, quiero construir un cuartel y reclutar soldados	I	I	P	P

User Persona 4 (Usuario inexperimentado)				
Historia de usuario	Independencia una de otras	Claridad	Relevancia para los clientes	Verificables
Como jugador sin experiencia, quiero un tutorial interactivo inicial	P	P	P	P
Como jugador social, quiero tener partidas multijugador rápidas	I	P	P	P
Como jugador, quiero recolectar recursos manualmente interactuando con los nodos	I	I	P	P
Como jugador, quiero acceder a un menú donde pueda ver a mis soldados reclutados	P	I	P	P

Tablas de la RTF hecha al grupo Feudalia en el repositorio de GitHub

Revisor: Feudalia

Revisado Historias de usuario

1. Crear un documento unificado con todas las Historias de Usuario
2. Mantener todas las historias desde el enfoque del usuario, es decir, asegurarse de que cada historia esté escrita desde el punto de vista del jugador o usuario final, como indica el estándar de control de calidad.
3. Definir previamente los tipos de usuario, crear perfiles coherentes, como: Usuario social Usuario competitivo Usuario novato Usuario experto
4. Añadir criterios de aceptación claros y verificables a todas las historias
5. Mantener separado en el backlog las historias de usuario, tareas técnicas, subtareas y bugs
6. Enlazar cada historia desde el Kanban a su documento correspondiente
7. Evitar subjetividades y requisitos no verificables

A partir de las RTF recibidas por parte de los otros grupos, hemos aplicado algunas recomendaciones, tales como:

3.3. Revisión de la gestión de riesgos

Hemos actualizado las medidas de gestión de los riesgos 1, 2, 3, 6, 7, añadiendo más detalles sobre el plan que seguiríamos en caso de que se produzca el riesgo.

Riesgo 1:

Anterior gestión: “En caso de que un miembro no pueda alcanzar el nivel requerido, redistribuir su carga hacia otros miembros y centrar su participación en áreas de menor complejidad técnica.”

Gestión actualizada: “Si después de 1 semana del último sprint, un miembro no puede alcanzar el nivel requerido, se redistribuirán sus tareas críticas a otros miembros y se le asignarán tareas de menor complejidad técnica, tales como documentación, testing, etc.”

Riesgo 2:

Anterior gestión: “Si el sistema cooperativo no se logra implementar a tiempo, ofrecer una versión alternativa offline o de turnos locales que mantenga la jugabilidad esencial.”

Gestión actualizada: “Si para el sprint 4 el sistema cooperativo no se logra implementar a tiempo, se activará el plan alternativo: ofrecer un modo local, offline, por turnos en un mismo dispositivo, manteniendo la jugabilidad base pero eliminando la conectividad online.”

Riesgo 3:

Anterior gestión: “Si se detecta retraso significativo, reducir el alcance del proyecto eliminando funcionalidades secundarias o simplificando características sin comprometer la funcionalidad principal, priorizando funciones clasificadas como Must”

Gestión actualizada: “Si se detecta retraso significativo (superior al 30% en cualquier sprint), se eliminarán 3 funcionalidades secundarias (2 Could y 1 Should), seleccionadas por votación del equipo basándose en el valor que aporta para el producto, todas las Won't y se eliminarán características que no comprometan la funcionalidad principal, priorizando funciones clasificadas como Must. Además, el equipo dedicará 3 horas extra cada semana hasta recuperar el cronograma.”

Riesgo 6:

Anterior gestión: “Si se detecta una desviación considerable, reestructurar el backlog priorizando funcionalidades "Must" y aplazando tareas "Should" o "Won't".”

Gestión actualizada: Si en dos sprints consecutivos se detecta una desviación considerable (<70% completitud), se reestructurará el backlog priorizando funcionalidades "Must", se aplazarán las tareas Should y Could menos prioritarias y se eliminarán las Won't.

Riesgo 7:

Anterior gestión: Si al finalizar un sprint muchas de las tareas han quedado inacabadas por falta de tiempo debido a exámenes, se establecerá para el siguiente sprint la cantidad proporcional de tareas hechas, basándonos en el tamaño adjudicado a las mismas.

Gestión actualizada: Durante periodos de exámenes, la carga de trabajo se reducirá al 50%, enfocándose únicamente en resolver deuda técnica y bugs críticos, sin desarrollar ninguna nueva funcionalidad que no sea _Must_. Si el retraso que provoca este riesgo es significativo, se implementará la gestión del riesgo 3.3

4. Descripción del trabajo realizado por cada miembro del grupo.

Además de todas las tareas realizadas en grupo y bajo consenso grupal, como las historias de usuarios, los plannings, retrospectivas, revisiones, etc, cada uno de nosotros ha realizado las siguientes tareas. Entre paréntesis se muestra el nombre de usuario de GitHub de cada integrante del grupo.

Daniel Pulido Leo (Danielpl-0):

- Creada la arquitectura base de la web (los tres documentos iniciales en HTML, JS y CSS).
- Desarrollada la primera interfaz gráfica de la web.
- Añadidos nuevos campos a rellenar a la hora de crear un perfil: foto, hábitos, intereses románticos y personalidad.
- Implementación del juego de dos verdades y una mentira.
- Limitados la altura y el peso que puede introducir un usuario.
- Introducida la opción de subir una segunda foto, y hacerla obligatoria.
- Añadir datos que faltaban a los 100 perfiles creados con el script.

Rodrigo Sánchez Peña (snchhz123):

- Arreglar el guardar del userID.
- Arreglar la lista de usuarios (Junto con Lucas Fernández-Galiano Catalán).
- Definir MVP, el producto mínimo viable.
- Definir la arquitectura inicial de la página web.
- Tener un botón en la pantalla de inicio que cuando se apriete muestre la lista de usuarios guardados en el servidor. (Junto con Lucas Fernández-Galiano Catalán).
- Arreglar el error que ponía “usuario conectado” dos veces encima del chat.

Alvaro Fu (alvaroo226):

- Escrita la página principal del GitHub y gran parte de la wiki, sobre todo el apartado de Home.
- Creado y escrito el documento README.
- Implementada la encuesta para la configuración de un perfil.
- Investigación y programación de pruebas unitarias (Junto con David Tijerín Antón).
- Cambiada la lista de usuarios para mostrar el género de cada usuario, en vez de sus gustos e intereses.
- Añadido campo de género a cada usuario.
- Desarrollado el sistema de puntuación entre dos usuarios al completar partidas (Junto con David Tijerín Antón).

David Tijerín Antón (DTA-4):

- Implementación del sistema de registro e inicio de sesión en la página web inicial.
- Sistema de chat grupal en el primer prototipo de página web.
- Creación del servidor de Firebase y traslado de las funcionalidades de la página web anterior.
- Sistema de chat individual en vez de grupal.
- Desarrollo e implementación del juego del 4 en raya.
- Integración del 4 en raya en el chat.

- Servidor en imgBB para las imágenes de la página web.
- Ajustes en los ficheros css para arreglar scrolls incorrectos y dar forma a los botones.
- Investigación y programación de pruebas unitarias (Junto con Alvaro Fu).
- Separación de los easter eggs a un archivo independiente
- Botón de usuario eliminado de la pantalla de inicio y logo actualizado.
- Añadir la opción de iniciar sesión con nombre de usuario y contraseña.
- Foto obligatoria al iniciar sesión y opciones de altura y peso añadidas.
- Desarrollado el sistema de puntuación entre dos usuarios al completar partidas (Junto con Alvaro Fu).
- Corrección de errores como destello en los chats o ajustes como mover funcionalidad entre botones.

Lucas Fernández-Galiano Catalán (lucassf10):

- Creado el apartado de “Gestión de Riesgos” en la wiki (Junto con David Tijerín Antón).
- Integrado el botón de “Descubre Gente” (anteriormente “Lista de Usuarios”).
- Añadido atributo de gustos e intereses a cada usuario.
- Desarrollado los filtros utilizados por la web en el apartado de Descubre Gente (comprobación de compatibilidad mutua entre géneros y gustos e intereses, que no haya ya chat abierto, que no estén bloqueados entre sí...).
- Añadida la persistencia de la sesión iniciada al refrescar la página web.
- Agregada la funcionalidad de eliminar chats.
- Implementada la opción de bloqueo y desbloqueo de usuarios.

Pablo Jofre Gómez (pablojofregomez):

- Creado la descripción y los criterios de aceptación de las historias de usuario (Junto con David Tijerín Antón).
- Hecho el principal diseño y desarrollo de la interfaz gráfica (paleta de colores, logo, pantalla de inicio, pantallas de error, *easter eggs*...).
- Adaptado la interfaz gráfica para que se vea igual en todos los dispositivos.
- Añadido el botón de “Ver Perfil” para poder ver más información de cada usuario.
- Rediseñado la lista de usuarios a una versión más orientada a usuarios, mostrando las fotos de perfil en grande junto con otros datos de interés.
- Vinculado la página web a un dominio y mantenimiento del mismo.
- Implementado el sistema de inicio de sesión por correo electrónico y contraseña
- Inyectado cien usuarios de prueba al sistema.
- Mejorado el sistema de gestión de carga de fotos.
- Añadido el apartado de “Puntos Amorosos” en la ventana de chat entre dos usuarios.
- Página de error 404.
- Introducir diversos easter eggs en la página web.

5. Código.

Para el proyecto hemos decidido crear una página web programada en html para los datos del navegador, css para los estilos y javascript para las funcionalidades. Así, el proyecto está dividido en los siguientes archivos:

- Para la página web principal:
 - index.html
 - Funciones globales y chat:
 - style.css
 - [app.js](#)
 - profile-styles.css (para la ventana de ver y editar perfil)
 - Juegos:
 - cuatroEnRaya.css
 - [cuatroEnRayaScript.js](#)
 - dosVerdades.css
 - [dosVerdades.js](#)
 - [gamesMenu.js](#) (para la funcionalidad del desplegable que deja elegir entre ambos juegos)
 - Ventana de descubre gente:
 - listausuarios.js
 - Easter eggs:
 - [easterEggs.js](#)
 - Pruebas unitarias:
 - [app.test.js](#)
 - [cuatroEnRayaScript.test.js](#)
 - [dosVerdades.test.js](#)
 - [jest.setup.js](#) (configuración de las pruebas unitarias)
- Para la página de error:
 - 404.html
 - style.css

Conformando a 12 de diciembre de 2025 los archivos javascript el 72,3% del código total, mientras que los css y html representan respectivamente el 20,1% y 7,6%

Todos ellos tienen una funcionalidad principal en la página web, excepto aquellos utilizados para las pruebas unitarias. Debido a que la página web se basa principalmente en el apartado visual, no se han podido realizar pruebas unitarias que englobasen todas las funcionalidades, pues muchas necesitaban de acciones específicas del usuario. Además, el uso de servicios externos como Firebase o imgBB también ha dificultado notablemente la tarea, pues son entornos difíciles de replicar a la hora de crear las pruebas. Por ello, hemos decidido escoger algunas de las funciones que considerábamos más importantes y locales dentro de los archivos javascript, y hemos programado pruebas unitarias para dichas funciones. En ellas hemos intentado englobar tanto los casos más normales como algunos más extremos, para asegurar el correcto funcionamiento de los scripts bajo cualquier condición. Se han realizado pruebas unitarias sobre las siguientes funciones:

- Sobre [app.js](#):

- generateChatId: genera el id del chat en Firebase con ciertas restricciones para que sea simétrico e identificable en un futuro y no haya errores de repetición.
- Sobre [cuatroEnRayaScript.js](#):
 - flatTo2D: convierte el tablero de fichas de un vector lineal (para guardarlo en Firebase) a una matriz que trata el programa.
 - board2DToFlat: convierte la matriz que representa el tablero a un vector
 - checkWin: comprueba si alguno de los dos jugadores ha hecho cuatro en raya en vertical, horizontal o diagonal.
 - checkTie: comprueba si todas las fichas están colocadas pero ninguno de los dos jugadores ha hecho cuatro en raya.
- Sobre [dosVerdades.js](#):
 - sendDosVerdades: envía las dos verdades al chat para que pueda verlas el otro usuario.
 - renderDosVerdadesFromMessage: muestra el mensaje con las tres frases en el chat.
 - submitGuess: lógica aplicada cuando el otro usuario elige qué frase en la mentira.

Así se ve la evolución que han sufrido las pruebas unitarias a lo largo del proyecto. En un principio nos costó mucho configurarlas y no entendíamos la forma ni de programarlas ni ejecutarlas consistentemente. Nos dieron muchos errores, pero no de funcionamiento, sino entre los scripts porque no accedían bien a los métodos sobre los que se aplicaban las pruebas. Esto se tradujo en que para cuando conseguimos programar una prueba unitaria en [app.js](#), llegamos a un estado del proyecto en el que nada de lo que íbamos a hacer afectaría a las funciones de ese script. Por ello, decidimos centrarnos en las funciones nuevas que estábamos programando en el momento. Así se ve cómo los archivos [cuatroEnRayaScript.js](#) y [dosVerdades.js](#) tienen una mayor cantidad de pruebas unitarias implementadas sobre los métodos más centrales en su funcionamiento.

En cuanto a la ejecución de las pruebas unitarias, utilizamos la extensión de java de Jest para configurarlas, y las ejecutábamos desde PowerShell. Además, solo vimos necesario que un miembro del grupo se instalase dicha extensión, ya que solo es necesaria para ejecutar las pruebas unitarias. Así lo que hacíamos era que cuando alguien modificaba algo del proyecto esa persona hacía pull desde GitHub y ejecutaba las pruebas unitarias en su ordenador para ver que todo lo anterior seguía funcionando correctamente.

Sin embargo, como se ha mencionado anteriormente la mayoría de pruebas del proyecto no se han podido programar debido a la alta carga de interacción con el usuario que requerían. Para ello creamos una carpeta pruebas en el repositorio, donde documentábamos todas aquellas pruebas manuales que veíamos más complejas y que necesitaban especificación. También es verdad que en el caso de estar reunidos, en vez de apuntar los requisitos en el documento simplemente se los comentábamos a la persona encargada de realizar los tests.