

Actividad Scrum

1. Un cliente quiere desarrollar una nueva red social de juegos de mesa y para ello se ha puesto en contacto con nosotros.

¿Qué va a querer el cliente?

El cliente solicita el desarrollo de una aplicación móvil compatible con iOS y Android que funcione como una red social centrada en juegos de mesa y cuyo acceso requiera autenticación de usuario. Al iniciar sesión, se presentará un muro social donde se visualizarán otros usuarios con sus fotos de perfil y posibilidad de chat con ellas, integrando además un sistema de geolocalización que, mediante previa autorización, permitirá organizar y mostrar partidas que se discutan en las cercanías del usuario.

Para aquellos contactos que formen parte de la lista de amigos, la interfaz dispondrá de un cuadro lateral específico para ellos que mostrara información detallada sobre los clubes a los que pertenecen y los torneos o juegos en los que se encuentran activos.

La plataforma informará sobre partidas físicas que se estén disputando en tiempo real junto con datos técnicos del juego en cuestión y permitirá la creación de blogs con texto e imágenes con posibilidad de contar en las publicaciones, y el contenido de estas moderados automáticamente por inteligencia artificial para asegurar el cumplimiento de las normas.

En un segundo nivel de prioridad, la aplicación debe ofrecer un registro completo de clubes, torneos y ligas disponibles con herramientas para comunicarse con estas entidades y la posibilidad de que los usuarios creen sus propias competiciones a cambio de una transacción económica. Esta sección incluirá un gestor integral de partidas, ya sean sueltas o parte de campeonatos, que mostrará la duración estimada o real, los participantes, los rankings, los sistemas de puntuación y los premios si existieran.

Finalmente, el proyecto contempla una fase futura para la implementación de un simulador de mesa digital que permita a los usuarios jugar a cualquier título directamente desde la aplicación conservando todas las funcionalidades sociales y de gestión previamente descritas.

2. El cliente se va a reunir con un miembro del equipo.

La tarea de esta persona consistirá en tomar nota de lo que el cliente necesita para ajustar el producto final lo más posible a la idea del cliente.

¿Con qué miembro del equipo se reunirá?

El cliente se reúne con el Product Owner (PO). Es la persona encargada de maximizar el valor del producto. Actúa como el interlocutor principal, traduciendo las necesidades del cliente en objetivos claros para el equipo de desarrollo.

¿Cuál será el resultado de esta reunión?

El resultado es el Product Backlog inicial.

Características principales:

- **Es una lista ordenada por prioridad.**
- **Contiene todas las funcionalidades, requisitos y mejoras deseadas (como el chat, la geolocalización o el muro).**
- **Es un documento vivo que evolucionará según cambien las necesidades del proyecto.**

3. Lo siguiente será realizar una división del producto, con el objetivo de desarrollar el Product Backlog:

ID	Usuario	Acción	Justificación	Estimación (Poker)	Prioridad (MoSCoW)	Prioridad (V/E)	Sprint
US01	Usuario	Autenticarse	Pueda acceder a mi perfil seguro	3	Must	Alta/Bajo	1
US02	Usuario	Ver un muro social	Pueda ver fotos y perfiles	5	Must	Alta/Medio	1
US03	Usuario	Chatear con otros	Pueda coordinar partidas	8	Must	Alta/Medio	1
US04	Usuario	Usar geolocalización	Encuentre partidas cerca de mí	13	Must	Alta/Alto	1
US05	Amigo	Ver cuadro lateral	Sepa en qué clubes están mis amigos	5	Should	Media/Bajo	2
US06	Usuario	Ver partidas en tiempo real	Conozca los datos técnicos del juego	8	Should	Media/Medio	2

US07	Usuario	Crear un blog con IA	Comparta contenido seguro y moderado	13	Should	Media/Alto	2
US08	Club/Org	Registrar torneos/ligas	Centralice la información competitiva	8	Could	Baja/Medio	Backlog
US09	Usuario	Crear competiciones	Organice mis propios eventos	20	Could	Baja/Alto	Backlog
US010	Jugador	Gestionar puntuaciones	Tenga un ranking y sistema de premios	13	Should	Media/Medio	2

Justificación de resultados de prioridad:

- **MoSCoW:** Se centra en la necesidad del negocio (sin Auth o Muro no hay aplicación).
- **Valor/Esfuerzo:** Identifica "Quick Wins". Por ejemplo, US05 da mucho valor y es poco esfuerzo, por lo que sube en la lista de tareas. Los resultados difieren porque MoSCoW es más estratégico y V/E es más táctico.

4. En nuestro proceso de planning póker (<https://planningpokeronline.com/>) vamos a utilizar una baraja que tiene las siguientes cartas 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100 y además, dos tarjetas: una con signo de interrogación (?) y otra con signo de café (∞).

El procedimiento será el siguiente:

- 1. Un moderador, que no jugará, preside la reunión apoyado y asesorado por el Gestor del Proyecto.**
- 2. El desarrollador con más conocimiento de una determinada característica proporciona una breve introducción sobre la misma. El equipo tiene la oportunidad de hacer preguntas y discutir para aclarar los supuestos y riesgos.**
- 3. Cada persona coloca una tarjeta boca abajo que representa su estimación. Las unidades utilizadas pueden ser variadas y definidas previamente. Pueden ser días de duración, días ideales o puntos de la historia. Durante el debate, los números no deben ser mencionados en absoluto.**
- 4. Todo el mundo muestra sus tarjetas de forma simultánea.**
- 5. A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa.**
- 6. Se vuelve a realizar el mismo proceso.**
- 7. Se suma la puntuación nueva y se añade como estimación de tiempo.**

5. También deberemos definir las prioridades de cada tarea y el sprint al que pertenecerá. ¿De cuánto tiempo será como mínimo y como máximo un Sprint? Aquí hay una serie de técnicas para priorizar tareas:

<https://www.atsistemas.com/blog/tcnicas-de-priorizacin-el-desafio-de-conseguir-un-orden-para-las-funcionalidades>

1. Definición del Tiempo de Sprint:

Según el marco de trabajo Scrum:

- **Mínimo: 1 semana.**
- **Máximo: 4 semanas (1 mes natural).**

Nuestra elección: Hemos fijado Sprints de 2 semanas. Esta duración permite tener suficiente tiempo para desarrollar funcionalidades completas (como la integración de mapas) sin perder la capacidad de pivotar rápido si el feedback del cliente cambia.

2. Justificación de los resultados de Prioridad:

Es común que MoSCoW y la Matriz Valor/Esfuerzo den resultados distintos. Aquí explicamos el por qué:

- **Enfoque MoSCoW: Se basa en la viabilidad del producto. Por ejemplo, el Chat (US03) es un "Must" porque una red social sin comunicación no funciona, independientemente de si es difícil o fácil de programar.**
- **Enfoque Valor vs. Esfuerzo: Se basa en el Retorno de Inversión. Aquí, el Cuadro Lateral de Amigos (US05) destaca porque ofrece mucho valor al usuario y es técnicamente sencillo. En un entorno puramente técnico, esta tarea podría saltar por delante de otras más complejas.**

Conclusión del proceso: Hemos decidido que el Sprint 1 se centre en los "Must" de MoSCoW para tener un Producto Mínimo Viable (MVP), mientras que en el Sprint 2 introduciremos las tareas que equilibran mejor el valor aportado con el esfuerzo requerido (como la US05 y US10).

3. Proceso de Trabajo:

- **Sesión de Poker:** Nos reunimos y cada uno lanzó su carta. Si había mucha diferencia (ej. alguien decía 3 y otro 13), los extremos explicaban su visión hasta llegar a un consenso.
- **Clasificación MoSCoW:** El Product Owner definió qué era innegociable para el lanzamiento inicial.
- **Mapeo de Esfuerzo:** El equipo de desarrollo comparó los puntos de Poker con el valor percibido para identificar los "Quick Wins".
- **Asignación de Sprint:** Cruzamos ambos resultados para llenar los Sprints sin sobrepasar nuestra capacidad de puntos por equipo (Velocity estimada: 30 puntos por Sprint).

6. ¡Por fin comenzamos el desarrollo! Comenzaremos con una reunión de planificación del sprint ¿cómo se llama? Una de las actividades que debemos hacer es dividir el Sprint en actividades más pequeñas (divide y vencerás)

¿Cómo se llama esta reunión? Se denomina Sprint Planning. En ella, el equipo decide qué historias de usuario se compromete a completar y cómo las llevará a cabo.

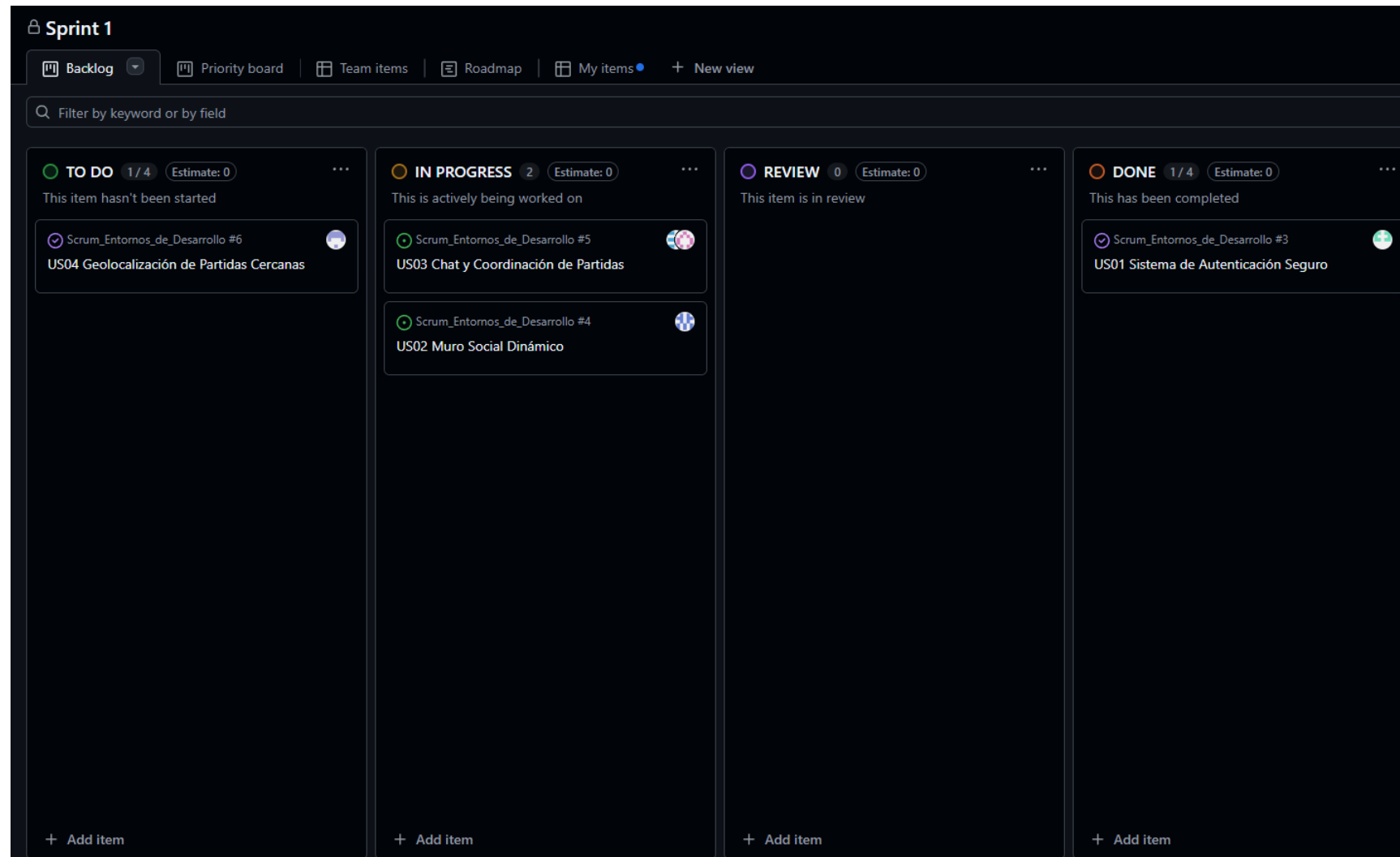
Una de las actividades críticas durante esta reunión es descomponer las Historias de Usuario seleccionadas en Tareas Técnicas o Sub-tareas.

Ejemplo de división para la US04 (Geolocalización):

- **Configurar API de Google Maps/Apple Maps.**
- **Desarrollar servicio de obtención de coordenadas del usuario.**
- **Diseñar la interfaz de "Partidas cercanas".**
- **Implementar la lógica de filtrado por radio de distancia.**

Esta división permite que varios desarrolladores trabajen en la misma historia de forma paralela y facilita el seguimiento del progreso diario.

7. El resultado del sprint planning va a ser el Sprint Backlog. Para su gestión podemos utilizar la técnica de Kanban. ¿Cómo quedaría un posible tablero?



“Ejemplo de un Kanban durante el desarrollo del primer Sprint”

8. Por si no te habías dado cuenta, antes te he engañado, no hemos comenzado todavía el desarrollo. Pero ahora sí que sí, tenemos nuestro panel Kanban y vamos a comenzar con las tareas de desarrollar. Vamos a ver que podría pasar durante el desarrollo.

Durante el desarrollo, las tarjetas del tablero Kanban se empiezan a mover de la columna "TO DO" a la de "IN PROGRESS". Lo que suele pasar en el día a día es lo siguiente:

- **Charlas rápidas cada mañana:** El equipo se reúne unos minutos para contar en qué está trabajando cada uno, asegurar que nadie está haciendo lo mismo y ver si alguien necesita ayuda.
- **Aparición de dudas:** Al ponerse a fabricar algo (como el chat o el mapa), surgen preguntas de detalle que no se vieron al principio. El equipo habla con el responsable del producto para aclarar cómo debe funcionar exactamente.
- **Imprevistos técnicos:** Es muy común que algo que parecía fácil se complique (por ejemplo, que el localizador no sea preciso en algunos móviles).
- **Apoyo entre compañeros:** Si un miembro del equipo termina su tarea antes, ayuda a los demás para que todas las funciones del grupo lleguen a tiempo al final de la etapa.
- **Comprobaciones:** Antes de dar algo por terminado, se revisa que funcione correctamente y que sea fácil de usar para que no tenga fallos cuando se le enseñe al cliente

9. Imaginemos que hemos terminado el primer Sprint, ¿qué toca ahora? Dos reuniones más:

- **Revisión del Sprint:**

Se muestra al cliente el trabajo que ya está listo. Esto sirve para que el cliente vea el progreso en “vivo” y nos diga qué le parece y nos dé ideas para mejorar la aplicación en los siguientes Sprints.

- **Retrospectiva del Sprint:**

Es una reunión interna del equipo para hablar sobre cómo han trabajado juntos. El objetivo no es mirar la aplicación, sino ver qué procesos han funcionado bien y cuáles han sido una molestia, para así buscar soluciones y trabajar más cómodos en el próximo Sprint.

10. ¿Y ahora qué?

Después de terminar las reuniones finales, el proceso no se detiene, sino que vuelve a empezar de forma mejorada:

- **El ciclo se reinicia:** El equipo vuelve al punto de partida para preparar la siguiente etapa de trabajo (el próximo Sprint).
- **Actualización de la lista:** El responsable del producto pone al día la lista de tareas. Añade las nuevas ideas que pidió el cliente en la reunión de revisión y quita o cambia lo que ya no es necesario.
- **Nueva planificación:** El equipo se reúne otra vez para elegir qué tareas de esa lista actualizada va a realizar en las próximas dos semanas.
- **Mejora constante:** Como en la reunión anterior hablaron de cómo trabajar mejor, ahora aplican esos cambios para ser más eficientes y evitar los errores que cometieron antes.