



Entregable 1.

Definición del roadmap para SolidarianID

Gestión Ágil de Proyectos

Máster Universitario en Ingeniería del Software

Autores:

Hernán Salambay Roldán
Alejandro Montoya Toro
Pedro Nicolás Gomariz
Aurora Hervás López
Dongyue Yu

10 de Noviembre de 2024



Facultad
de Informática
UMU



Contenidos

1. Introducción	3
2. Resolución del trabajo	3
2.1 Visión	3
2.1.1 Por qué estamos aquí	3
2.1.2 Elevator pitch	3
2.1.3 Caja del producto	4
2.1.4 Lista de NOes	4
2.1.5 Vecinos	5
2.1.6 Mostrar la solución	6
2.1.7 Qué nos quita el sueño (riesgos)	7
2.1.8 Tamaño	7
2.1.9 Alcance vs. Tiempo vs. Coste vs. Calidad	8
2.1.10 Coste	8
2.2 Roadmap	9
2.2.1 Describir procedimiento para priorizar	9
2.2.2 Pronóstico de velocidad	10
2.2.3 Plan de versiones	11
2.3 Versión 1: MVP	12
2.3.1 Describir condiciones de satisfacción de la release (DoD)	12
2.3.2 Describir procedimiento para priorizar	13
2.3.3 Plan de iteraciones	15
2.4 Iteración 1	15
2.4.1 Definición de hecho (DoD)	16
2.4.2 Tablero Scrum	17
2.4.3 Sprint Backlog	18
2.4.3.1 Objetivo del Sprint	21
2.4.3.2 Mecanismo de Priorización	22
2.4.4 Criterios de aceptación	22
3. Conclusiones	27
4. Bibliografía	27



1. Introducción

Este documento sobre el proyecto SolidarianID está estructurado para proporcionar una descripción detallada de cada aspecto relevante en el desarrollo y despliegue del Minimum Viable Product (MVP). Su organización busca ofrecer una visión clara y comprensible de los objetivos, alcance y planificación del proyecto así como de las decisiones clave.

La resolución del trabajo comienza con el análisis de la visión del producto, describiendo su propósito y objetivos generales, y se analizan aspectos clave como el equilibrio entre alcance, tiempo y coste. También se identifican riesgos potenciales y se establecen las prioridades y limitaciones que guiarán el desarrollo del proyecto.

Seguidamente, se presenta un roadmap que orienta la evolución del producto, con los planes de desarrollo, priorización y lanzamientos, lo cual es fundamental para guiar la ejecución de las fases de desarrollo en un marco ágil. Finalmente, se especifican las condiciones para el lanzamiento del MVP y las primeras iteraciones del proyecto, que establecen la base para alcanzar la satisfacción del cliente y asegurar una entrega funcional y escalable.

Finalmente, el documento cierra con un análisis de resultados y las conclusiones obtenidas, seguidas de una bibliografía que respalda el desarrollo y la planificación del proyecto.

2. Resolución del trabajo

2.1 Visión

En esta sección analizaremos los distintos aspectos que conforman la visión del proyecto, tales como el por qué estamos aquí, el elevator pitch, los riesgos, y otros elementos esenciales para contextualizar y orientar el desarrollo de SolidarianID. Para esto, empleamos la técnica *Agile Inception Deck*, que mediante una serie de preguntas clave guía la alineación del equipo y los stakeholders en torno a la misión, objetivos y límites del proyecto.

El objetivo del *Agile Inception Deck* es proporcionar una base compartida de entendimiento que permita a todos los involucrados comprender hacia dónde se dirige el desarrollo del producto y asegurar que las decisiones estratégicas están orientadas al valor y las necesidades del usuario final.

2.1.1 Por qué estamos aquí

Queremos construir una plataforma que permita a personas de todo el mundo participar y apoyar causas solidarias de forma sencilla, facilitando la creación de comunidades y la promoción de causas a través de un sistema accesible, seguro e internacionalizado.

2.1.2 Elevator pitch

SolidarianID es una plataforma que une a ciudadanos de todo el mundo para impulsar acciones solidarias. A través de comunidades, los usuarios pueden apoyar causas, hacer donaciones y ofrecer su tiempo como voluntarios. A diferencia de otras plataformas solidarias, SolidarianID ofrece un ID único para cada usuario, lo que proporciona transparencia y un registro claro de sus acciones en apoyo a diferentes causas. Además, pueden participar en eventos, seguir a otras personas y generar conexiones significativas. Juntos, estamos construyendo un espacio donde la solidaridad y el compromiso social pueden florecer. ¡Únete a nosotros y transforma tu deseo de ayudar en acción!



2.1.3 Caja del producto

En este apartado se establecen las características únicas y beneficios del producto, persuadiendo al cliente potencial para que lo considere como una opción viable.



Figura 1. Caja del producto de SolidarianID.

Las principales características y beneficios de SolidarianID son las siguientes:

- **Identidad Digital Única:** Cada usuario registrado obtiene un ID único que rastrea su historial de participación, incluyendo comunidades, causas apoyadas y acciones solidarias, promoviendo la transparencia y el reconocimiento.
- **Construcción de Comunidades:** Los usuarios pueden crear y unirse a comunidades enfocadas en causas específicas, fomentando la interacción, colaboración y el intercambio de ideas entre personas con intereses similares.
- **Mecanismo de Donación y Participación:** La plataforma permite a los usuarios realizar donaciones y participar en acciones solidarias, facilitando la movilización de recursos y el apoyo a iniciativas concretas que generan un impacto positivo.
- **Impulso a la Solidaridad:** Facilita la participación en causas sociales, permitiendo a los usuarios hacer una diferencia tangible en sus comunidades y en un ámbito global.

2.1.4 Lista de NOes

En este apartado se define la lista de NOes que ayuda a definir los límites y principios éticos que SolidarianID debe seguir para garantizar que cumpla su misión de manera efectiva y responsable.

A continuación, se listan algunas de las cosas que se deben evitar en el desarrollo y operación de SolidarianID:



- **No a la Falta de Transparencia:** no se debe ocultar información sobre el uso de fondos o el impacto de las donaciones.
- **No a la Exclusividad:** no permitir el acceso limitado a comunidades o causas, asegurando que todos puedan participar.
- **No al Uso Inadecuado de Datos:** no manejar datos personales sin el consentimiento informado de los usuarios, debe cumplirse la normativa de protección de datos.
- **No a la Publicidad Engañosa:** no hacer promesas o afirmaciones exageradas sobre el impacto de las donaciones o la participación.
- **No a la Exclusión de ODS:** no permitir causas que no se alineen con los Objetivos de Desarrollo Sostenible (ODS).
- **No a la desviación de fondos:** no usar el dinero recaudado para fines distintos a las causas solidarias.

2.1.5 Vecinos

Nuestra plataforma busca situarse en el centro de un ecosistema diverso de agentes vecinos que trabajan por el bien social, formando una red de posibles colaboradores y participantes que fortalecerán su desarrollo, alcance e impacto.

Entre los principales vecinos de SolidarianID podemos encontrar:

- **ONGs y Fundaciones:** Organizaciones dedicadas a causas sociales y ambientales, que podrían aprovechar la plataforma para atraer voluntarios y captar fondos, promoviendo su labor de manera más visible y accesible.
- **Instituciones Educativas:** Universidades, colegios y centros de formación interesados en proyectos de servicio comunitario y sensibilización social.
- **Empresas y Corporaciones Tecnológicas y Financieras:** Las organizaciones con programas de Responsabilidad Social Corporativa (RSC) pueden ver SolidarianID como una herramienta para canalizar sus iniciativas de voluntariado y donaciones, además de aportar al desarrollo de la plataforma desde el punto de vista tecnológico y financiero.
- **Agentes de la Comunidad y Voluntarios:** Ciudadanos y líderes comunitarios que ya participan en el apoyo social en sus entornos. SolidarianID les ofrece un espacio para ampliar su impacto, conectar con otras comunidades y dar visibilidad a sus causas locales.
- **Redes Sociales y Medios de Comunicación:** Canales de difusión que juegan un rol importante al potenciar la visibilidad de las causas promovidas en SolidarianID, ayudando a difundir las campañas y eventos en sus plataformas.
- **Gobiernos y Agencias Públicas:** Instituciones y administraciones que promueven el bienestar social y que, a través de SolidarianID, pueden acercarse a la ciudadanía para sumar esfuerzos en iniciativas solidarias, proporcionando recursos o aprobando y apoyando actividades.
- **Otras Plataformas Solidarias o de Crowdfunding:** Plataformas como GoFundMe y Change.org, que también trabajan en la recaudación de fondos y el activismo social. SolidarianID podría colaborar con estas plataformas en causas comunes.

Estos agentes vecinos pueden aportarnos valor en las distintas etapas del uso de la plataforma, desde la difusión de causas y captación de fondos hasta la organización de actividades y voluntariados.

2.1.6 Mostrar la solución

La plataforma se basará en una arquitectura robusta y escalable, diseñada para facilitar la interacción entre usuarios y permitir la incorporación ágil de futuras mejoras. Este sistema modular responde de manera flexible y eficiente a las necesidades de los usuarios.

A continuación, se detallan los **componentes clave** que componen esta solución:

- **Backend:** Desarrollado en **Node.js** y **NestJS**, con una arquitectura de microservicios organizada según principios de Arquitectura Hexagonal y Domain Driven Design (DDD) para maximizar la modularidad y facilidad de mantenimiento.
- **Frontend:** Implementado en **React** para una interfaz intuitiva y adaptable en dispositivos móviles y de escritorio, garantizando una experiencia de usuario fluida.
- **API y Conectividad:** La plataforma ofrecerá una API documentada con **OpenAPI** y consultas con **GraphQL**, permitiendo integración con otras aplicaciones y acceso a datos de manera precisa y eficiente.
- **Aplicación Móvil:** Desarrollada en **React Native** para proporcionar una experiencia similar a la de aplicaciones nativas en **iOS** y **Android**.
- **Seguridad y Autenticación:** Para proteger la información y el acceso, SolidarianID usará **OAuth2** y **JWT**, que aseguran tanto la autenticación de usuarios como el control de acceso seguro.
- **Almacenamiento de Datos:** Combinación de bases de datos relacionales y NoSQL, lo que permite un rendimiento optimizado y almacenamiento seguro adaptado a cada tipo de necesidad dentro de la plataforma.
- **Integración Continua y Despliegue:** Un pipeline CI/CD basado en la nube nos permitirá despliegues frecuentes y seguros, garantizando que SolidarianID esté siempre disponible y funcional para los usuarios.

A continuación se muestra un diagrama de arquitectura que muestra una versión sencilla de como quedaría la solución:

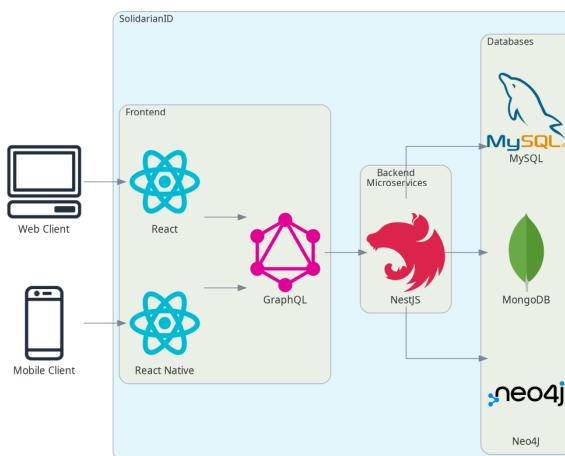


Figura 2. Diagrama de arquitectura simplificado de SolidarianID.



2.1.7 Qué nos quita el sueño (riesgos)

El riesgo es un factor externo que afecta de manera negativa a un proyecto reduciendo sus posibilidades de éxito. Es necesario realizar un ejercicio para descubrir todos los posibles riesgos a los que puede estar expuesto el proyecto, ya que de esta manera somos conscientes de ellos e intentamos mejorar el proyecto todo lo posible para reducir sus probabilidades de ocurrencia.

Los riesgos identificados han sido los siguientes:

- **Baja adopción de la plataforma:** puede darse el caso que una vez lanzada la aplicación al mercado no resulte tan exitosa como se esperaba.
- **Posibles estafas que se puedan dar dentro de la plataforma:** algunos usuarios pueden crear comunidades falsas y definir acciones solidarias dentro de ellas con el único objetivo de lucrarse económica mente a costa de la solidaridad del resto de usuarios de la plataforma.
- **Problemas de seguridad y privacidad:** aparición de ciberdelincuentes que consigan romper los mecanismos de seguridad y acceder a los datos privados y sensibles de los usuarios.
- **Problemas asociados a no cumplir alguna legislación vigente:** no cumplir con alguna de las leyes nacionales sobre las TICs durante el desarrollo y despliegue de la plataforma.
- **Problemas de estabilidad de la plataforma:** se cae con regularidad debido a fallos internos por una mala codificación, o bien no es capaz de aguantar a una gran cantidad de usuarios conectados simultáneamente.
- **Falta de colaboración de ONGs:** una vez lanzada la plataforma, las ONGs no se implican ni contribuyen en su popularización.
- **Aparición de otras plataformas similares en el mercado:** antes o después del lanzamiento de la aplicación al mercado, podrían aparecer otras aplicaciones similares en el mercado que absorvieran a los potenciales clientes.

2.1.8 Tamaño

En este apartado se pretende realizar una estimación preliminar y de alto nivel sobre el esfuerzo, recursos y tiempo que el proyecto podría requerir. La idea es ofrecer una idea aproximada de la magnitud y complejidad del producto para que los stakeholders comprendan la escala de lo que se desea construir, sin entrar en un cálculo preciso o exhaustivo.

Para poder definir el tamaño de las Hisotrias de Usuario (HUs), es necesario establecer la duración del Sprint. En nuestro caso, hemos establecido una duración de 2 semanas para cada uno de los Sprint, que hemos considerado tiempo suficiente para asegurar una entrega constante de valor dentro de este proyecto. Aplicando la técnica *T-Shirt Sizing Agile Project* tenemos las siguientes tallas con su duración:

Talla	XS	S	M	L	XL
Duración	-1 Sprint	1-2 Sprint	3-4 Sprint	4-6 Sprint	+6 Sprint



Para asignar tallas a las nuestras HUs, partimos de la de ‘Registro de usuario’, que no es ni la más pequeña ni la más grande. Asignamos a esta historia una talla S y tomándola como referencia definimos las demás. Asignado las tallas anteriores a un subconjunto de las HUs obtenemos la siguiente tabla:

Épica	Historias de usuario	Talla
Gestión de usuarios	Registro de usuario	S
	Login de usuario	XS
Gestión de comunidades	Creación de una comunidad	S
	Explorar comunidades disponibles	XS
Gestión de causas solidarias	Apoyar una causa solidaria	XS
Gestión de acciones solidarias	Contribuir a una acción solidaria	S

2.1.9 Alcance vs. Tiempo vs. Coste vs. Calidad

El principal objetivo del proyecto es desarrollar una plataforma global que facilite la participación ciudadana en causas solidarias y promueva la interacción entre voluntarios. El alcance del proyecto incluye: funcionalidad para el registro de usuarios, para crear, explorar y unirse a comunidades, la creación de una interfaz intuitiva y funcionalidad para permitir que los usuarios den apoyo y contribuyan a causas solidarias.

En este contexto, la **calidad** es la prioridad principal. Se desea construir una plataforma de calidad desde el principio, realizando pruebas y obteniendo retroalimentación constante durante el desarrollo para asegurar que funcione de manera efectiva y segura y evitar errores a largo plazo.

Por otra parte, el **tiempo** es un factor crítico, ya que el proyecto debe completarse en 8 meses. Para asegurar que se cumplan los plazos y asegurar el progreso, se establecerán hitos claros y se mantendrá consistencia entre las iteraciones.

En cuanto al **alcance**, es crucial ajustarlo para que se adapte al tiempo y coste disponibles. Puesto que hay una ambición considerable en la funcionalidad que se desea incluir en la plataforma, se priorizan las características esenciales y se dejará el resto para futuras iteraciones, asegurando que el producto inicial sea funcional.

Por último, el **coste**, aunque es un factor importante, es menos prioritario que la calidad, el tiempo y el alcance, ya que se puede ajustar para garantizar que se cumplan las expectativas de calidad y funcionalidad y no se puede determinar con exactitud desde el inicio.

2.1.10 Coste

Para calcular el coste del proyecto, se deben considerar varios factores que contribuyen al presupuesto total. El principal componente del coste viene determinado por los recursos humanos. El desarrollo completo de la plataforma, desde el backend hasta el frontend, diseño de la interfaz, seguridad y DevOps, será realizado por un equipo de 5 ingenieros. Considerando que todos los componentes del equipo reciben el mismo salario, alrededor de



1.800 euros al mes, y que el tiempo estimado de desarrollo coincide con la duración del máster, unos 8 meses (de octubre a mayo), el coste en cuanto a recursos humanos se estima en 72.000 euros.

Por otro lado, se consideran otros costes asociados al proyecto, que incluyen posibles gastos en licencias de software, infraestructura tecnológica y marketing. Estos costes podrían variar con el proyecto pero se puede considerar un presupuesto de unos 18.000 euros para asegurar un funcionamiento estable durante el primer año. Por último, se destinarán 5.000 euros a una campaña de marketing inicial centrada en redes sociales para dar a conocer la plataforma y atraer usuarios.

Si sumamos los costes de recursos humanos, infraestructura y marketing, tenemos que el coste final del proyecto es de **95.000** euros.

2.2 Roadmap

En esta sección, presentaremos el roadmap del proyecto, el cual detalla las etapas y hitos clave en el desarrollo de SolidarianID. El roadmap sirve como una guía estratégica para alcanzar nuestra visión, alineando los esfuerzos del equipo y facilitando la colaboración con los stakeholders. Al desglosar las funcionalidades y entregables en un marco temporal, aseguramos que todos los involucrados tengan claridad sobre las prioridades, plazos y objetivos a corto y largo plazo.

Para la planificación, hemos implementado la técnica de *User Story Mapping* en Miro ([enlace al tablero](#)), lo que nos permite visualizar las necesidades y experiencias de nuestros usuarios de manera colaborativa y estructurada. A través del mapeo de HUs, se han desglosado las funcionalidades del producto en tareas más manejables para poder priorizarlas según su valor e impacto. Este enfoque fomenta una comprensión compartida entre todos los miembros del equipo, y también asegura que nuestro desarrollo esté alineado con las expectativas de los usuarios finales, manteniendo un enfoque ágil en el desarrollo de SolidarianID.

2.2.1 Describir procedimiento para priorizar

En el tablero Miro se definen diferentes roles que consideramos necesarios para abordar las funcionalidades:

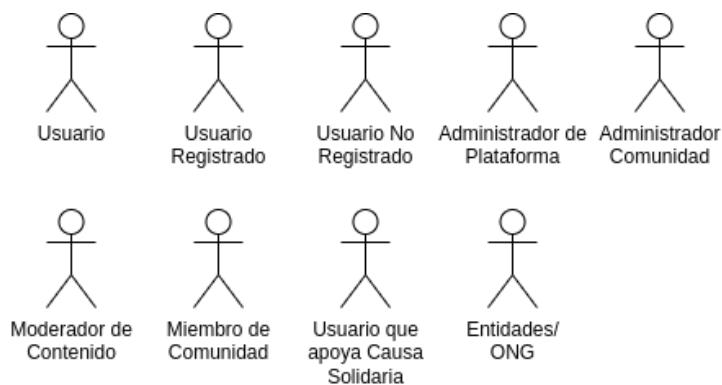


Figura 3. Diagrama de actores para SolidarianID.

A partir de estos roles, se ha completado el flujo narrativo indicando las épicas que engloban las HUs que tengan un objetivo común. Se ha seguido la técnica *MoSCoW* para asignar prioridad a las historias de usuario. Según esta técnica, los niveles de prioridad se clasifican en **Must have** (debe tener), **Should have** (debería tener), **Could have** (podría tener) y **Won't have**



but would like to in the future (no tendrá pero me gustaría en el futuro). Esta técnica nos ayuda a establecer qué se debe hacer primero, qué se debe hacer después y qué no se necesita hacer y así poder planificar las iteraciones del proyecto.

En nuestro caso, se han marcado como *M* todas las HUs que se deben implementar para el MVP, que son las que se consideran más importantes para ofrecer la funcionalidad básica en la primera versión. Finalmente, todas aquellas funcionalidades marcadas con una *M* y algunas con una *S*, han sido desglosadas en una serie de subtareas o alternativas de implementación. Aquellas marcadas con *M* serán las más prioritarias y por tanto deberían realizarse primero, no obstante, si disponemos de tiempo de sobra sería interesante incluir la funcionalidades marcadas con *S* también para el MVP.

2.2.2 Pronóstico de velocidad

El pronóstico de velocidad es una técnica utilizada para estimar cuánto trabajo puede completar un equipo de desarrollo durante una iteración o sprint. La velocidad se refiere a la cantidad de trabajo que el equipo ha logrado completar en iteraciones anteriores, convirtiéndose en una medida clara del ritmo de progreso del equipo.

En ausencia de un histórico de velocidad, hemos decidido utilizar un enfoque de pronóstico. Para ello, descomponemos las historias de usuario en tareas más específicas, estimando la cantidad de horas necesarias para cada tarea y calculando cuántas de estas historias podrán abordarse en la primera iteración. Para ello, hemos llegado al acuerdo de aplicar la siguiente fórmula:

$$1 \text{ semana} = 5 \text{ días} \times 5 \text{ horas/día} \times 5 \text{ desarrolladores} = 125 \text{ horas}$$

Aunque una jornada laboral estándar es de 8 horas diarias, hemos ajustado el cálculo a 5 horas de trabajo productivo diario. Este ajuste considera que el equipo no dedica las 8 horas exclusivamente al desarrollo, ya que parte del tiempo se invierte en reuniones, comunicación (emails, mensajería), y otras interrupciones propias de la jornada. Este enfoque nos proporciona una estimación más precisa de las horas realmente disponibles para el trabajo de desarrollo.

Aplicando el resultado anterior y considerando que un sprint tiene una duración de 2 semanas, tenemos como resultado 250 horas para realizar las tareas de la primera iteración.

$$1 \text{ sprint} = 2 \text{ semanas} \times 125 \text{ horas} = 250 \text{ horas}$$

Partimos de la HU “*Editar perfil*” con un tamaño de camiseta de *S* como base para realizar la estimación en horas para las demás tareas:

	SOL-100	Diseño de la arquitectura de la funcionalidad	4 h
	SOL-102	Diseño de la interfaz de usuario	6 h
	SOL-103	Implementación del frontend	5 h
	SOL-104	Implementación del backend	5 h
	SOL-105	Implementación de las pruebas	4 h
	SOL-106	Documentación	2 h

Figura 4. Estimación temporal de subtareas del proceso de registro de un usuario.



En la siguiente tabla, se muestra la combinación de las estimaciones de horas con las tallas de camiseta, utilizando la secuencia de Fibonacci como base para reflejar el esfuerzo relativo de cada tarea.

Talla	Horas estimadas	Esfuerzo
XS	10	1
S	26	2
M	35	3
L	45	5
XL	60	8

Después de definir las tallas de camiseta con sus estimaciones en horas y esfuerzo pasaremos a la estimación de nuestro MVP. Partiendo de las informaciones anteriores calculamos el tiempo en semanas para realizar la entrega del producto:



Figura 5. HU de cada talla para el MVP.

Estimación de MVP = $(15 \times 10 \times 1) + (9 \times 26 \times 2) + (2 \times 35 \times 3) / 125$ horas semanales = 828 horas / 125 horas semanales = **6,624 semanas ≈ 7 semanas**.

Dado el análisis de las tareas y las horas estimadas, podemos concluir que necesitamos al menos **4 sprints** para realizar una entrega con valor que cumpla con los requisitos básicos del producto.

2.2.3 Plan de versiones

En cuanto al Plan de versiones, la primera versión a entregar corresponde con el MVP, que incluye la funcionalidad básica de la plataforma integrada en una aplicación web adaptable, desarrollada durante el primer cuatrimestre. A partir de esta versión, se seguirá añadiendo funcionalidad y desarrollando próximas versiones. En la segunda versión se ampliará la interfaz para mejorar la experiencia de usuario.

El desarrollo se estructurará en una serie de versiones, cada una de las cuales introducirá funcionalidades clave de forma incremental, siguiendo el enfoque de MVP para facilitar el feedback temprano y la mejora continua.

A continuación, se describen las versiones planificadas:



Fecha	(15/01/2025)	Periodo 2	Periodo 3
Nombre	MVP	V2	V3
Objetivo	Lanzar una versión base funcional que lleve a la realidad el concepto de la plataforma	Expansión de Funcionalidades y API Pública	Aplicación Móvil y Experiencia Multiplataforma
Funcionalidades	<ul style="list-style-type: none"> • Registro y autenticación de usuarios. • Perfil personal y público de usuarios. • Seguimiento de usuarios • Creación y gestión esencial de comunidades. • Creación y gestión de causas y acciones solidarias. • Mecanismo básico de donación. • Informes de comunidades • Estadísticas de uso de plataforma 	Abrir la plataforma para colaborar con ONGs y entidades externas mediante una API pública, mejorar las herramientas de gestión para administradores y ampliación y mejora de la interfaz.	Llevar SolidarianID a dispositivos móviles con una aplicación para iOS y Android, mejorando la experiencia de uso.
Métricas de éxito	<ul style="list-style-type: none"> • Los usuarios pueden realizar acciones solidarias e interactuar con comunidades. • Los administradores de la plataforma y de las comunidades pueden hacer gestiones esenciales de la plataforma y las comunidades respectivamente. 	Los usuarios y organizaciones podrán disfrutar de funcionalidades avanzadas.	Los usuarios podrán acceder a la aplicación desde diversas plataformas.

2.3 Versión 1: MVP

El Producto Mínimo Viable (MVP) consiste en una versión del producto con las suficientes características para satisfacer a los primeros clientes y obtener retroalimentación para el posterior desarrollo de la plataforma.

Para desarrollar el MVP de SolidarianID, hemos usado la plataforma de Jira Software ([enlace al proyecto](#)) para definir las épicas, historias de usuario y las subtareas que abarca la primera versión.

En esta sección se describen las condiciones de satisfacción de la release (DoD), así como el procedimiento seguido para estimar el esfuerzo de las historias de usuario, establecer su prioridad y un plan de iteraciones.

2.3.1 Describir condiciones de satisfacción de la release (DoD)

Para un MVP, el Definition of Done (DoD) se debe centrar en cubrir las funcionalidades esenciales que demuestran el valor del producto. A continuación se muestran el conjunto de requisitos que deben cumplirse para que la versión se considere terminada y entregada:

- **Funcionalidades Imprescindibles:** todas las funcionalidades acordadas para esta versión cumplen con los criterios de aceptación y funcionan según lo esperado.
- **Interfaz de Usuario:** la interfaz es clara, intuitiva y accesible, permitiendo una navegación fluida que facilita la experiencia del usuario.



- **Testing:** se han realizado pruebas unitarias en todas las funciones clave con al menos el 80% de cobertura de código. Además, se realizarán pruebas de integración entre los módulos principales de la aplicación.
- **Documentación:** se aportará un documento el cual, al menos deberá incluir una descripción básica del sistema destacando las decisiones de diseño más relevantes. Además se adjuntará diagramas estructurales que faciliten la comprensión del sistema.
- **Rendimiento:** las pruebas de carga iniciales asegurarán que el sistema soporta un nivel básico de usuarios simultáneos sin fallos significativos.
- **Seguridad:** se deberán implementar los mecanismos mínimos de seguridad básicos que garanticen la protección de los datos.
- **Despliegue:** el MVP deberá ser accesible a través de un entorno de producción estable para los usuarios finales.
- **Aprobación del Product Owner:** el PO validará que el sistema está alineado con las especificaciones acordadas y que cumple con la lógica de negocio.

2.3.2 Describir procedimiento para priorizar

El valor de cada HU será asignado por el Product Owner, el cual conoce las necesidades del cliente y puede dar una estimación mucho más acertada. Para ello, tendrá en cuenta diferentes métricas de cada HU, como los beneficios, el riesgo de implementar o valor frente a la competencia. El rango de valores que utilizará se encuentra entre el 1 y el 10, donde el 1 indica la HU con más prioridad y el 10 con menos prioridad.

Para definir el esfuerzo de cada HU, hemos utilizado la técnica de *Puntos de Historia*, a través de la cual se asignan valores de esfuerzos a historias haciendo uso de una secuencia de valores concreta elegida. En nuestro caso hemos optado por la secuencia de *Fibonacci*, la cual es una de las más utilizadas. Como punto de partida para poder aplicar la técnica, es necesario elegir dos HU las cuales consideremos que tengan un valor de esfuerzo de 2 y 5, las cuales serán nuestras historias de referencia. A partir de ellas, ya podremos seguir eligiendo HU dentro de las disponibles para el MVP y asignarle un valor dentro de la secuencia seleccionada, utilizando como referencia las dos historias anteriores para saber si suponen un esfuerzo mayor o menos a estas.

Las historias de referencia elegidas han sido '*Visualización del perfil*', a la cual le hemos asignado un valor de esfuerzo de 2, y '*Editar perfil*', a la que hemos asignado un valor de 5.

Como el equipo de desarrollo se compone de más de 1 persona, es necesario llegar a un consenso en la asignación de valores de esfuerzo. Para ello hemos empleado la técnica *Planning Poker* a través de la aplicación web Planning Poker Online ([enlace](#)). Allí, hemos ido mostrando diferentes HU y cada miembro del equipo le ha asignado un valor de esfuerzo según su consideración personal. Cuando todos han votado, se han mostrado en pantalla las asignaciones de cada miembro del equipo, y en caso de discrepancias entre valores, se ha llegado a un consenso.

En la siguiente imagen, utilizando la aplicación, cada miembro del equipo de desarrollo ha elegido un valor de esfuerzo para la HU '*Login de usuario*'. Como podemos ver, ha habido diferencias entre los valores, ya que Pedro ha asignado un valor de 2, mientras que el resto de miembros han seleccionado el valor 3.

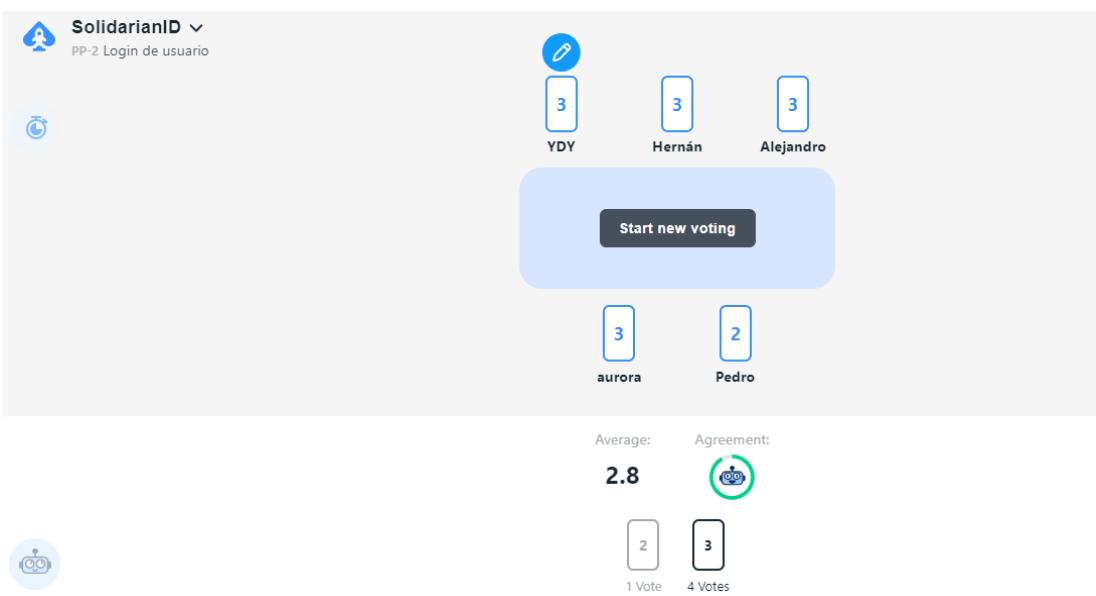


Figura 6. Primera ejecución del Planning Poker para la HU *Login de usuario*.

Tras esto, se ha abierto un debate dentro del equipo donde se ha llegado a un consenso sobre la estimación del esfuerzo de la HU. Se ha vuelto a votar y se han obtenido los siguientes resultados.

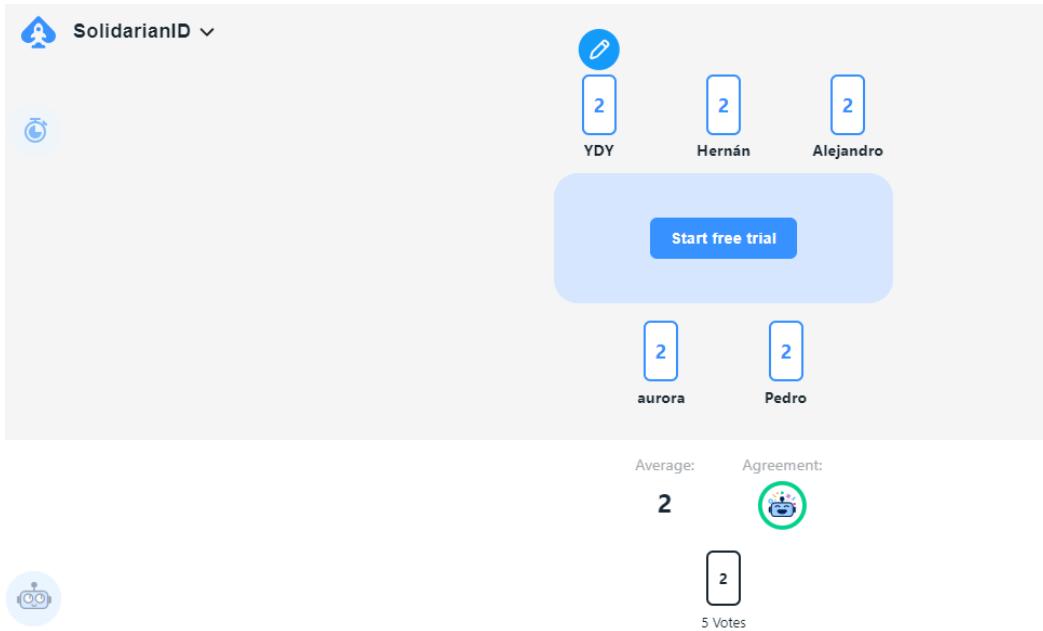


Figura 7. Segunda ejecución del Planning Poker para la HU *Login de usuario*.

Este proceso se ha seguido realizando hasta que se ha asignado un valor de estimación de esfuerzo para cada una de las HU definidas para el MVP.

Finalmente, el criterio utilizado para priorizar ha sido la relación entre el valor de una HU y su esfuerzo, cuya fórmula se detalla a continuación:

$$\text{Prioridad de HU} = \text{Valor de HU} / \text{Esfuerzo de HU}$$



Una vez obtenidos los valores de prioridad, se han ordenado las HU en función de ese valor de manera decreciente y se ha procedido a planificar los sprint o iteraciones del proyecto.

2.3.3 Plan de iteraciones

En este apartado se detallan las iteraciones para elaborar la primera versión de SolidarianID:

Iteración	HUs	Puntos de historia	Objetivo
1	Registro de usuario, Login de usuario, Editar perfil, Visualizar perfil público del usuario, Crear comunidad, Dar de alta una causa solidaria, Ver listado de comunidad, Ver detalle de una comunidad, Validar creación de comunidad, Ver listado de causas solidarias, Ver detalle de una causa solidaria, Unirse a una comunidad.	38	Implementar las funcionalidades básicas para establecer el uso y navegación principal de la plataforma
2	Búsqueda de una comunidad, Consultar historial personal, Seguir usuario, Consultar historial de usuario de la comunidad, Crear acciones solidarias, Ver listado de acciones solidarias, Ver detalle de una acción solidaria, Apoyar una acción solidaria, Admitir usuario a la comunidad.	≈ 38	Implementa las funcionalidades que facilitan la interacción y participación de los usuarios en comunidades y causas, promoviendo el compromiso activo.
3	Búsqueda de causas solidarias, Búsqueda de una acción solidaria, Contribuir a una acción solidaria, Ver estadísticas del uso de la plataforma, Visualizar informe	≈ 38	Implementa las funcionalidades que ayudan a mejorar la experiencia del usuario con métricas, notificaciones y opciones avanzadas, aumentando la visibilidad del impacto.

Partiendo de las informaciones del apartado **2.2.2 Pronóstico de velocidad** obtenemos que en cada iteración estimamos un trabajo de 38 puntos de historia completando en torno a 10 historias de usuarios.

2.4 Iteración 1

En esta sección se describe la planificación de la primera iteración, que coincide con la primera parte del MVP. En esta iteración, el equipo se dedicará a diseñar los primeros elementos del backend y una interfaz de usuario básica. Durante esta fase se configurarán las herramientas de desarrollo y se sentarán las bases para asegurar un desarrollo incremental y orientado a los objetivos principales de SolidarianID.



A continuación, se detalla la DoD, el tablero Scrum y el Sprint Backlog, que se han realizado en el proyecto Jira del equipo, para gestionar las tareas seleccionadas y los Criterios de Aceptación para las historias de usuario.

2.4.1 Definición de hecho (DoD)

La DoD es un conjunto de criterios que una tarea debe cumplir para considerarse completa y lista para ser entregada. Estos criterios aseguran no solo que el equipo mantenga ciertos estándares de calidad y consistencia en las entregas, sino que también permite tener entre todos los miembros una visión compartida de lo que implica terminar una tarea.

Criterios generales de la DoD:

- El código relacionado con la tarea debe estar escrito, probado y funcionar como se espera.
- La tarea debe incluir pruebas unitarias y de integración que aseguren el comportamiento esperado.
- La tarea cumple con los requisitos funcionales especificados en la HU y los criterios de aceptación definidos.
- El código se ajusta a la Coding Style Guide definida (convención de nombres, formateo, comentarios y estructura de archivos y carpetas).
- El código sigue los principios de Clean Architecture y los patrones de diseño del enfoque DDD.
- Se incluye la documentación necesaria para facilitar su compresión y mantenimiento en el futuro.
- La funcionalidad de la tarea se debe integrar adecuadamente en el repositorio correspondiente al proyecto.

Consideraciones no funcionales:

- Toda interfaz visible al usuario y mensajes de error deben estar preparados para ser traducidos fácilmente.
- La tarea debe cumplir con los requisitos básicos de accesibilidad (WCAG 2.2 AA).
- El código debe estar optimizado para evitar tiempos de respuesta altos, garantizando una experiencia fluida.
- Se implementarán las medidas de seguridad básicas aplicables, alineadas al menos con el OWASP Top Ten.
- La tarea debe cumplir con las directrices de la Ley Orgánica 3/2018 de Protección de Datos Personales y garantía de los derechos digitales.

Dado que no todos los criterios se aplican a todas las tareas (por ejemplo, tareas de documentación o diseño), cada tarea debe evaluarse para determinar qué criterios son relevantes. A continuación, se definen algunos subcriterios específicos según el tipo de tarea:

- **Tareas de Documentación:**
 - La documentación debe ser clara, estar actualizada y accesible en el repositorio correspondiente.



- Siempre que sea posible se incluirán diagramas que faciliten la comprensión del diseño.

- **Tareas de Diseño:**

- Los diseños deben ser consistentes y encajar adecuadamente con el resto del proyecto.
- Los diseños relevantes, serán revisados al menos por otro miembro del equipo con experiencia suficiente.

- **Tareas de Configuración/Infraestructura:**

- Se debe asegurar que los cambios a la infraestructura de producción no afectan negativamente en el rendimiento (puede involucrar pruebas de carga o estrés).
- Se adoptarán las medidas de seguridad necesarias para garantizar la protección de la infraestructura (autenticaciones seguras, Principle of Least Privilege (PoLP), encriptación, etc...).

2.4.2 Tablero Scrum

En esta sección explicamos el tipo de tablero que usaremos para visualizar y gestionar las historias de usuario en un sprint. Para ello, hemos usado el tablero *Kanban* que proporciona la herramienta Jira. Hemos configurado este tablero con cuatro columnas que representan las distintas fases del flujo de trabajo:

- **Por hacer:** HU asignada al sprint que todavía no está iniciada. Una vez iniciada, se mueve a la columna *En progreso*.
- **En progreso:** HUs en las que el equipo está trabajando actualmente. Una vez terminadas, se pasan a la columna *En revisión*.
- **Bloqueado:** HUs que no pueden continuar avanzando y están temporalmente detenidas. Por ejemplo, tareas que están en espera de implementación de otras tareas. Una vez resuelto el obstáculo se puede pasar a la columna *En progreso* o *En revisión*.
- **En revisión:** HUs completadas a falta de revisión por otros miembros del equipo. Una vez revisadas y aprobadas pasarán a la columna *Listo*.
- **Listo:** HUs que han sido terminadas, revisadas y cumplen con los criterios de aceptación para ser desplegadas.

Figura 8. Tablero Kanban.

El tablero, además de permitir visualizar el flujo de trabajo de cada HUs permite ver quién es el responsable de ésta, que será el encargado de pasar de una etapa a otra.



La actualización en tiempo real del progreso de las HUs permitirá a todo el equipo mantener una visión clara del flujo de trabajo. Para reforzar esto, se llevará a cabo una reunión diaria de 15 minutos (*Daily Scrum* o *Stand-up*), en el cual cada miembro compartirá sus avances, los obstáculos encontrados y las posibles soluciones a ello. Durante esta reunión cualquier miembro del equipo puede crear, modificar o eliminar tareas según sea necesario, pero siempre con el consenso del equipo.

Al finalizar el sprint, se realizará una reunión de revisión denominada *Sprint Review*, donde el equipo compartirá el avance y los logros alcanzados en el sprint. Esta reunión permite evaluar el incremento del producto y recibir retroalimentación, fomentando la mejora continua en futuras iteraciones.

2.4.3 Sprint Backlog

Nuestro Sprint Backlog ([enlace](#)) contiene la lista de tareas que el equipo se ha comprometido a completar durante el primer sprint. En este caso, hemos seleccionado tareas a partir de las historias de usuario y épicas definidas en el backlog general, priorizando aquellas que aportan el mayor valor al MVP y que son fundamentales para el desarrollo inicial de la plataforma. Las historias de usuario se han ido eligiendo conforme al diagrama que se muestra a continuación, teniendo en cuenta que las horas disponibles del sprint sean suficientes para completar todas las tareas de cada HU hasta que no quepan más.

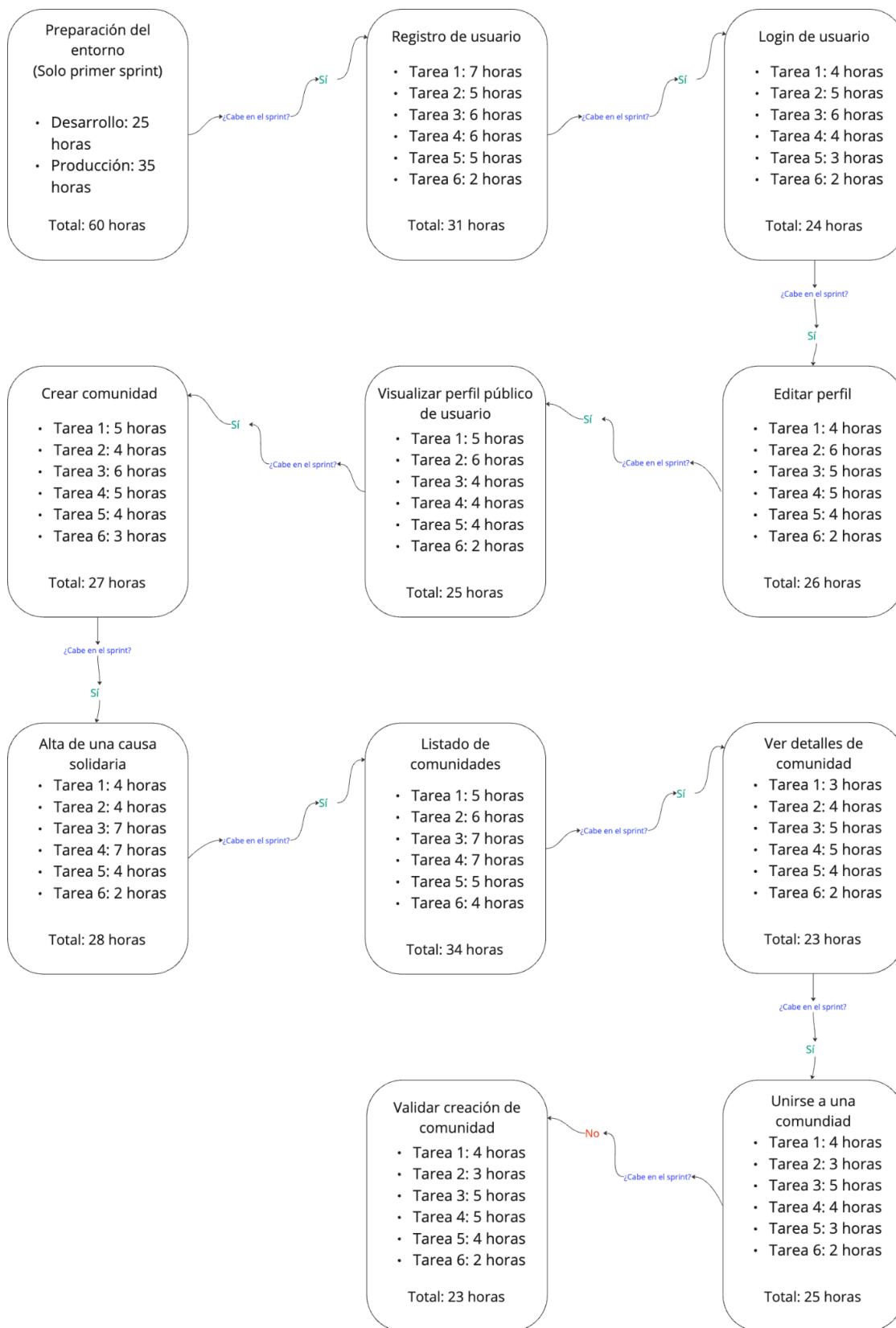


Figura 9: Diagrama de planificación del primer sprint.

A continuación, se muestra una tabla que refleja las HUs que hemos elegido para el primer sprint, detallando las estimaciones en horas para cada una de las subtareas de cada una de ellas.



Historia de usuario	Subtareas	Estimación en horas
Registro de usuario	Diseño de la arquitectura de la funcionalidad	7
	Diseño de la interfaz de usuario	5
	Implementación del frontend	6
	Implementación del backend	6
	Implementación de las pruebas	5
	Documentación	2
	Total	31
Login de usuario	Diseño de la arquitectura de la funcionalidad	4
	Diseño de la interfaz de usuario	5
	Implementación del frontend	6
	Implementación del backend	4
	Implementación de las pruebas	3
	Documentación	2
	Total	24
Editar perfil	Diseño de la arquitectura de la funcionalidad	4
	Diseño de la interfaz de usuario	6
	Implementación del frontend	5
	Implementación del backend	5
	Implementación de las pruebas	4
	Documentación	2
	Total	26
Visualizar perfil público del usuario	Diseño de la arquitectura de la funcionalidad	5
	Diseño de la interfaz de usuario	6
	Implementación del frontend	4
	Implementación del backend	4
	Implementación de las pruebas	4
	Documentación	2
	Total	25
Creación de una comunidad	Diseño de la arquitectura de la funcionalidad	5
	Diseño de la interfaz de usuario	4
	Implementación del frontend	6
	Implementación del backend	5
	Implementación de las pruebas	4
	Documentación	3
	Total	27
Dar de alta una causa solidaria	Diseño de la arquitectura de la funcionalidad	4



	Diseño de la interfaz de usuario	4
	Implementación del frontend	7
	Implementación del backend	7
	Implementación de las pruebas	4
	Documentación	2
	Total	28
Ver listado de comunidades	Diseño de la arquitectura de la funcionalidad	5
	Diseño de la interfaz de usuario	6
	Implementación del frontend	7
	Implementación del backend	7
	Implementación de las pruebas	5
	Documentación	4
	Total	34
Ver detalle de una comunidad	Diseño de la arquitectura de la funcionalidad	3
	Diseño de la interfaz de usuario	4
	Implementación del frontend	5
	Implementación del backend	5
	Implementación de las pruebas	4
	Documentación	2
	Total	23
Unirse a una comunidad	Diseño de la arquitectura de la funcionalidad	4
	Diseño de la interfaz de usuario	3
	Implementación del frontend	5
	Implementación del backend	4
	Implementación de las pruebas	3
	Documentación	2
	Total	21
Preparación del entorno de desarrollo	Total	25**
Despliegue en Producción	Total	35
	Total Sprint	248

**5 horas * 5 desarrolladores

2.4.3.1 Objetivo del Sprint

El objetivo de esta iteración es establecer una base funcional clara para el desarrollo de SolidarianID, en el cual nos enfocamos en las funcionalidades esenciales como el registro e



inicio de sesión de usuarios, la creación y gestión básica de comunidades, y el sistema inicial de gestión de donaciones y acciones solidarias.

2.4.3.2 Mecanismo de Priorización

Como hemos mencionado en apartados anteriores, para asignar prioridad a las historias de usuario en el sprint, se ha utilizado la técnica *MoSCoW*, enfocándonos en aquellas historias asignadas con M y S (*Must Have* y *Should Have*) que cumplen con los objetivos del MVP. A partir de estas historias, definimos y refinamos sus tareas correspondientes y, mediante la técnica de *Planning Poker* con la participación de todo el equipo, asignamos las estimaciones para cada una.

2.4.4 Criterios de aceptación

En esta sección describiremos los criterios de aceptación, que no son más que condiciones específicas y resultados mínimos que una historia de usuario o tarea debe cumplir para ser considerada completa y aceptada por el cliente o los stakeholders.

En esencia, un criterio de aceptación de calidad, establece el *qué* debe lograrse, definiendo claramente el comportamiento, funcionalidad o característica que debe estar presente para satisfacer los requisitos. Para ello utilizaremos el método *SMART* que un acrónimo que establece cinco características clave que los objetivos deben cumplir para maximizar sus probabilidades de éxito: deben ser **Especificos** (*Specific*), **Medibles** (*Measurable*), **Alcanzables** (*Achievable*), **Relevantes** (*Relevant*), y **Limitados en el tiempo** (*Time-bound*).

A continuación presentamos los criterios de aceptación para las historias de usuario que incluiremos en el primer sprint:

SOL-1: Registro de usuario

Como usuario no registrado **quiero** poder registrarme en la plataforma proporcionando mi nombre, teléfono móvil, ciudad y país **para** obtener un SolidarianID y poder participar en acciones solidarias

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario que quiere registrarse, **cuando** ingreso mi correo electrónico y una contraseña válida y hago clic en ‘Registrarse’, **entonces** mi cuenta se crea exitosamente y recibo un mensaje de confirmación indicando que el registro ha sido exitoso

- Escenario de fracaso:

Dado que soy un usuario que quiere registrarse, **cuando** ingreso mi correo electrónico y una contraseña válida y hago clic en ‘Registrarse’, **entonces** recibo un mensaje de error que indica que el correo ya está en uso



SOL-3: Login de usuario

Como usuario registrado **quiero** iniciar sesión en la plataforma,
para interactuar en la plataforma con mi perfil de usuario

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario registrado, **cuando** ingreso mi correo electrónico y contraseña correctamente y hago clic en ‘Iniciar sesión’, **entonces** puedo acceder a mi cuenta y se muestra la página de inicio

- Escenario de fracaso:

Dado que soy un usuario registrado, **cuando** ingreso mi correo electrónico y contraseña correctamente y hago clic en ‘Iniciar sesión’, **entonces** recibo un mensaje de error indicando que las credenciales son incorrectas y no puedo acceder a mi cuenta

SOL-96: Editar perfil

Como usuario registrado **quiero** editar mi perfil público **para** elegir la información que se muestra en la plataforma

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario registrado que quiere actualizar su información, **cuando** inicio sesión, accedo a mi perfil, realizo cambios en los datos o la visibilidad y hago clic en ‘Guardar cambios’, **entonces** mi perfil se actualiza correctamente.

- Escenario de fracaso:

Dado que soy un usuario registrado que quiere actualizar su información, **cuando** inicio sesión, accedo a mi perfil, introduzco datos no válidos y hago clic en ‘Guardar cambios’, **entonces** recibo un mensaje de error que indica que debo corregir los campos y no se actualiza mi perfil.



SOL-82: Visualizar perfil público del usuario

Como usuario, **quiero** visualizar el perfil público de otro usuario **para** poder obtener información sobre él.

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario que accede a la plataforma, **cuando** hago clic en el perfil de otro usuario, **entonces** puedo ver su perfil público.

- Escenario de fracaso:

Dado que soy un usuario que accede a la plataforma, **cuando** hago clic en el perfil de otro usuario, **entonces** recibo un mensaje de error indicando que se ha producido un error y no se muestra el perfil del usuario.

SOL-2: Creación de una comunidad

Como usuario registrado, **quiero** crear una comunidad introduciendo un nombre único y una descripción, **para** atraer a otros usuarios con intereses comunes y promover causas solidarias

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario registrado, **cuando** completo los campos de nombre y descripción y hago clic en ‘Crear comunidad’, **entonces** se crea la comunidad y soy asignado como administrador

- Escenario de fracaso:

Dado que soy un usuario registrado, **cuando** dejo algún campo obligatorio vacío y hago clic en ‘Crear comunidad’, **entonces** recibo un mensaje de error indicando que debo completar los campos correctamente



SOL- 4: Dar de alta una causa solidaria

Como administrador de una comunidad, **quiero** definir al menos una causa solidaria vinculada a la comunidad, **para** asegurar que la comunidad tiene un propósito claro y alineado con los Objetivos de Desarrollo Sostenible (ODS)

Criterio de aceptación

- Escenario exitoso:

Dado que soy un administrador de una comunidad, **cuando** completo el formulario para crear una causa solidaria vinculada a la comunidad y hago clic en ‘Crear causa’, **entonces** la causa se crea y queda visible en el listado de la comunidad

- Escenario de fracaso:

Dado que soy un administrador de una comunidad, **cuando** dejó algún campo vacío en el formulario para crear una causa solidaria vinculada a la comunidad y hago clic en ‘Crear causa’, **entonces** recibo un mensaje de error indicando que debo completar todos los campos

SOL-6: Ver listado de comunidades

Como usuario **quiero** explorar las comunidades disponibles en la plataforma **para** participar en las acciones solidarias

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario **cuando** entré en la plataforma, **entonces** puedo ver un listado de comunidades disponibles

- Escenario de fracaso:

Dado que soy un usuario **cuando** entré en la plataforma para ver las comunidades disponibles, **entonces** recibo un mensaje de error indicando que no hay comunidades



SOL-81: Ver detalles de una comunidad

Como usuario **quiero** ver detalles de una comunidad **para** poder obtener más información sobre ella

Criterio de aceptación

- Escenario exitoso:

Dado que soy un usuario que accede a la plataforma, **cuando** hago clic en una comunidad del listado, **entonces** puedo ver los detalles completos de la comunidad.

- Escenario de fracaso:

Dado que soy un usuario que accede a la plataforma, **cuando** hago clic en una comunidad del listado, **entonces** recibo un mensaje indicando que los detalles de la comunidad no están disponibles y no se muestra ninguna información adicional.

SOL- 5: Unirse a una comunidad

Criterio de aceptación

Como usuario registrado, **quiero** solicitar unirse a una comunidad existente **para** poder participar en sus causas solidarias

- Escenario exitoso:

Dado que soy un usuario registrado **que** ha iniciado sesión y está visualizando una comunidad, hago clic en ‘Unirse’ en una comunidad **entonces** me uno y mi nombre aparece en la lista de miembros de la comunidad

- Escenario de fracaso:

Dado que soy un usuario registrado **que** ha iniciado sesión y está visualizando una comunidad, hago clic en ‘Unirse’ en una comunidad **entonces** recibo un mensaje de error indicando que no se pudo llevar a cabo la unión



3. Conclusiones

En la realización de este entregable, algunas actividades resultaron más sencillas de completar que otras. La definición de visión de proyecto, apoyada en la técnica *Agile Inception Deck*, fue particularmente directa y eficiente. Esta herramienta nos proporcionó una estructura clara para establecer los objetivos y el propósito de SolidarianID, facilitando una alineación inmediata del equipo. Además, el uso de Miro para el mapeo de HUs nos permitió una organización rápida de las funcionalidades clave, haciendo que esta actividad se desarrollara con fluidez y agilidad.

Por otro lado, las actividades de estimación de esfuerzo y planificación de velocidad fueron más complicadas. La falta de un histórico de velocidad nos obligó a partir de cero, lo cual aumentó la dificultad de realizar predicciones precisas sobre el trabajo que podía completarse en cada iteración. Alcanzar un consenso durante el *Planning Poker* también fue un desafío, ya que distintos miembros del equipo teníamos percepciones distintas sobre el esfuerzo requerido para ciertas HUs. Esto añadió tiempo adicional a la actividad, haciendo que fuera una de las más demandantes.

Entre las actividades más formativas, podemos destacar la definición de la visión y la priorización de tareas mediante la técnica *MoSCoW*. Estas actividades nos permitieron profundizar en la comprensión de los objetivos y necesidades del proyecto.

En cambio, la estimación de costos fue una de las actividades menos formativas. Al tratarse de un análisis no orientado a negocio, no se han explorado posibles estrategias de financiamiento ni se han evaluado diferentes escenarios de inversión. Como resultado, esta actividad no nos ha aportado tanto en el desarrollo de habilidades prácticas.

Para finalizar, consideramos que este ha sido un trabajo altamente formativo, que nos ha permitido ampliar nuestro conocimiento en técnicas ágiles y planificación de proyectos. Además, nos ha permitido adquirir y comprender una base sólida para el desarrollo de SolidarianID, alineando al equipo en una visión compartida y estableciendo una estructura clara y funcional para el desarrollo del proyecto.

4. Bibliografía

- [1] Agile Inception: Qué es y cómo ejecutarlo.
- [2] Elevator pitch o cómo convencer a un inversor en 30 segundos.
- [3] Agile Project Initiation Techniques - The Inception Deck & Boot Camp, Jonathan Rasmusson.
- [4] ¿POR QUÉ tienes que usar UN INCEPTION? - Javier Garzás.
- [5] ¿POR QUÉ tienes que usar USER STORY MAPS? - Javier Garzás.
- [6] El método MoSCoW: priorización simple de tareas en un Proyecto.
- [7] Diapositivas de las sesiones 4 y 5 de Gestión Ágil de Proyectos.



Entregable 2.

Gestión ágil de equipos, riesgos, escalado y seguridad para SolidarianID

Gestión Ágil de Proyectos

Máster Universitario en Ingeniería del Software

Autores:

Hernán Salambay Roldán
Alejandro Montoya Toro
Pedro Nicolás Gomariz
Aurora Hervás López
Dongyue Yu

8 de Enero de 2025



Facultad
de Informática
UMU



Contenidos

1 Introducción	4
2 Resolución del trabajo	4
2.1 Seguimiento	4
2.2 Peopleware	6
2.2.1 Ice breaker - Personal Map	7
2.2.2 Motivado: Moving motivators	7
2.2.3 Motivado: Feedback wrap	8
2.2.4 Auto-organizado: Team barometer	11
2.2.5 Auto-organizado: Toma de decisiones: Apocalipsis zombie	12
2.2.6 Auto-organizado: Tableros de delegación	13
2.2.7 Auto-organizado: Matrices de multifuncionalidad	14
2.2.8 Mejora continua – Estrella de mar	15
2.2.9 Motivado: Calendarios Niko Niko	16
2.3 Riesgos	18
2.3.1 Metodología empleada	18
2.3.2 Identificación y Clasificación de Riesgos	18
2.3.3 Análisis de Riesgos	20
2.3.4 Planificación de la Gestión de Riesgos	21
2.3.5 Monitorización de Riesgos	22
2.3.6 Impediment Backlog Inicial	23
2.4 Escalado	23
2.4.1 Selección del modelo SAFe®	24
2.4.2 Plan de actuación del proyecto	25
2.4.3 Definición de PI Planning	27
2.4.4 Arquitectura de ART	29
2.4.4.1 Roles claves	30
2.4.4.2 Composición de los equipos del ART	30
2.4.4.3 Eventos del ART	31
2.4.4.3 Herramientas y Tecnologías	32
2.4.5 Identificación de flujos de valor	32
2.4.6 Escalado a Configuración de Portafolio	33
2.4.6.1 Épicas Estratégicas	33
2.4.6.2 Gestión de las Épicas con el Kanban de Portafolio	34
2.4.7 Resolución de una Épica	35
2.4.7.1 Creación de los Capabilities	35
2.4.7.2 Desglose en Features y Enablers	36
2.5 Seguridad	37
2.5.1 Amenazas SolidarianID, a nivel de gobernanza	37
2.5.2 DFD SolidarianID	39



2.5.3 Amenazas técnicas para SolidarianID	41
2.5.4 Secure Scrum para amenazas técnicas identificadas	43
3 Conclusiones	46
4 Bibliografía	47



1 Introducción

Este documento se compone de varias secciones, cada una relacionada con los temas que hemos ido aprendiendo en las sesiones de teoría y prácticas.

En primer lugar, se presenta el seguimiento ficticio de un proyecto utilizando JIRA, con un enfoque en la planificación y monitoreo del progreso. A continuación, se expone la aplicación de las metodologías de trabajo entre equipos, como Peopleware y Management 3.0, resaltando su influencia en la colaboración. Seguidamente, se analizan posibles riesgos y amenazas asociados al desarrollo de la futura aplicación SolidarianID.

Finalmente, se responden las preguntas planteadas y se incluyen reflexiones sobre el conocimiento adquirido durante el desarrollo de los proyectos y actividades, proporcionando una visión integral del proceso y los aprendizajes obtenidos.

2 Resolución del trabajo

2.1 Seguimiento

El seguimiento de un proyecto es el proceso continuo de monitorear, evaluar y registrar el progreso del mismo para asegurar que las tareas se completen según lo planificado, dentro del tiempo, presupuesto y alcance establecidos. Para ello, hemos usado la herramienta [Jira](#), diseñada para la gestión de proyectos y el seguimiento de tareas en equipo.

De acuerdo con nuestro roadmap, establecimos una disponibilidad de 5 horas efectivas diarias por persona para el desarrollo del proyecto. Esto equivale a un total de 25 horas diarias dedicadas al proyecto (5 miembros x 5 horas/día).

Dado que existe una discrepancia entre el desarrollo real del proyecto y la elaboración de este documento, hemos decidido realizar una simulación del seguimiento.

Durante la semana del 9 al 15 de diciembre, hemos accedido a nuestro proyecto en Jira, asignando horas de trabajo a las tareas correspondientes en nuestro Sprint Backlog. Al inicio del sprint, asignamos una historia de usuario a cada miembro del equipo. Cada miembro se hizo responsable de completar todas las subtareas incluidas en la historia que le fue asignada, asegurando así una distribución equitativa y clara del trabajo.

En la Figura 1 podemos observar las historias de usuario que hemos incluido en nuestro sprint (son historias de usuario para un sprint de dos semanas). Además, aparece el estado de las cinco primeras tareas, la persona asignada y los puntos de historia asignados.



The screenshot shows the Jira Sprint Backlog for 'EQUIPO 1' under 'Sprint 1'. It lists 10 user stories (incidencias) from 'SOL-94' to 'SOL-4'. Each story has sub-tasks and is categorized by epic and type. A detailed view of 'SOL-94' shows its sub-tasks: 'PREPARACIÓN DEL ENTORNO DE DESARROLLO' (EN PROGRESO), 'GESTIÓN DE USUARIOS' (EN PROGRESO), 'GESTIÓN DE COMUNIDADES' (TAREAS POR HACER), 'GESTIÓN EN ACCIONES SOLIDARIAS' (TAREAS POR HACER), and 'GESTIÓN DE CAUSAS SOLIDARIAS' (TAREAS POR HACER).

Figura 1. Sprint Backlog.

Durante el simulacro, gestionamos el avance de las historias de usuario a través del [tablero Kanban](#), moviéndolas progresivamente entre las columnas de “POR HACER”, “EN PROGRESO”, “BLOQUEADO”, “EN REVISIÓN” y “LISTO” como se puede ver en la Figura 2. Una historia de usuario se considera en estado “LISTO” únicamente cuando todas las subtareas asociadas han sido completadas y cumplen con sus criterios de aceptación.

The screenshot shows the Jira Kanban board for 'EQUIPO 1' under 'Sprint 1'. The board is divided into five columns: 'POR HACER 5', 'EN PROGRESO 4', 'BLOQUEADO', 'EN REVISIÓN', and 'LISTO 1'. Stories move from left to right as they are completed. In the 'Por Hacer' column, stories include 'Ver listado de comunidades', 'Unirse a una comunidad', 'Ver detalles de una comunidad', 'Crear comunidad', and 'Dar de alta una causa solidaria'. In the 'En Progreso' column, stories include 'Registro de usuario', 'Login de usuario', 'Visualizar perfil público del usuario', 'Editar perfil', and 'Preparación del entorno de desarrollo'. The 'Bloqueado' and 'En Revisión' columns are currently empty. The 'Listo' column contains one story, 'Preparación del entorno de desarrollo', which is fully completed (green checkmark).

Figura 2. Tablero Kanban Sprint 1.

A continuación, se muestra un diagrama, Figura 3, generado por el propio Jira que refleja el avance de las historias de usuario durante una semana de trabajo, que son cinco días laborales. El eje X representa los días de la semana y el eje Y, los puntos de historia. Podemos ver que inicialmente tenemos 34 puntos de historia y 10 historias de usuario. Durante los dos primeros días, no se termina ninguna historia de usuario, por lo que la línea roja se mantiene estable. En el tercer día podemos ver que la línea roja va bajando, indicando que hemos terminado algunas historias de usuario.

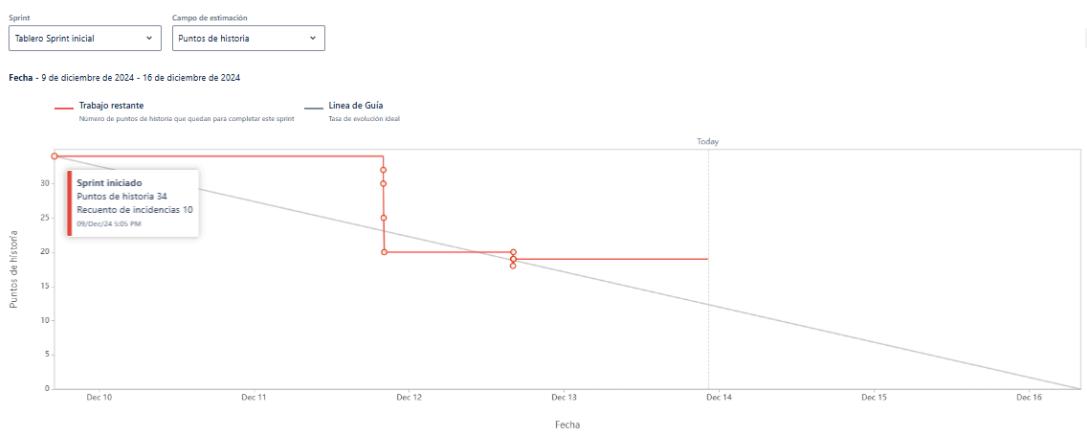


Figura 3. Diagrama burn down.

En el cuarto día, Figura 4, entró una incidencia nueva que no estaba en la planificación inicial. Por lo que aumentaron los puntos de historia. Podemos ver que ese mismo día también terminamos una historia de usuario.

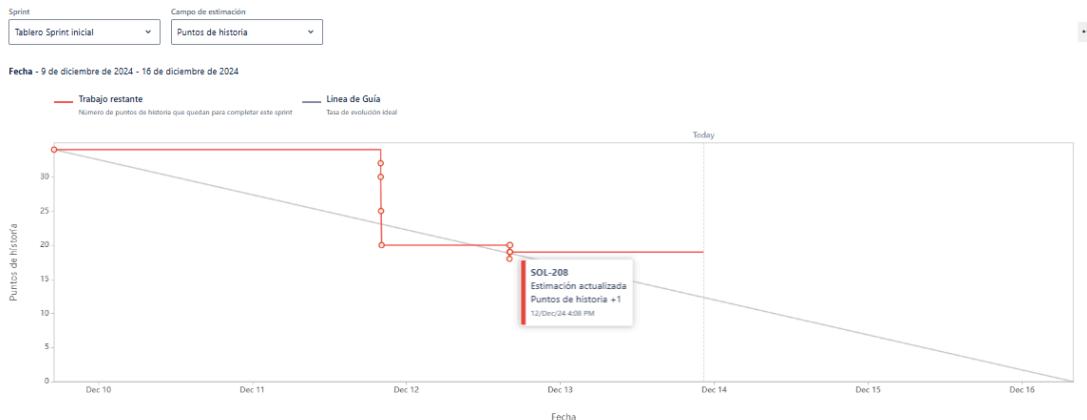


Figura 4. Diagrama burn down.

Finalmente, vemos que la línea roja está por encima de la línea gris, eso significa que en el primer sprint no se ha conseguido acabar todas las historias de usuarios escogidas. Este resultado es el esperado, ya que hemos escogido más historias de usuario de las que el equipo realmente puede cubrir.

2.2 Peopleware

El término *Peopleware* hace referencia a las personas como el elemento más importante en el desarrollo de proyectos, especialmente en entornos ágiles. Este enfoque destaca la importancia de los aspectos humanos, como la motivación, la comunicación y la colaboración, para el éxito de los equipos.

En este apartado, se abordan prácticas específicas de *Peopleware* aplicadas por nuestro equipo, como el uso de herramientas para fomentar la cohesión y el análisis de dinámicas internas, siguiendo principios de Management 3.0. Estas prácticas son esenciales para crear un ambiente de trabajo productivo y mantener un equipo comprometido y enfocado en los objetivos comunes.

Todas estas actividades se han llevado a cabo principalmente en el tablero [Sesión 7. Equipos ágiles 24-25](#) de Miro a lo largo de las sesiones de la asignatura, lo que nos ha permitido avanzar y aplicar de manera práctica todas las técnicas aprendidas en clase, fomentando una dinámica de trabajo colaborativa.

2.2.1 Ice breaker - Personal Map

Los [Personal Maps](#) son una herramienta visual y práctica utilizada en metodologías ágiles para romper el hielo y conocer mejor a los miembros de un equipo. Consisten en mapas mentales que permiten representar aspectos clave de la vida, intereses y motivaciones de cada persona y fortalecer las relaciones y la empatía dentro del equipo.

Para realizarlo, se parte del nombre de una persona en el centro del mapa y se dibujan ramas que reflejan distintos aspectos de su vida. En nuestro caso, las categorías utilizadas fueron: Familia, Infancia, Estudios, Superpoder, Valores, Objetivos y Aficiones. Cada integrante del grupo completó su mapa personal y presentó a otro compañero ante la audiencia, lo que permitió descubrir detalles interesantes y conectar de una manera más cercana. En la Figura 5 se muestran ejemplos de los mapas personales elaborados por el equipo. Finalmente, debatimos sobre los Personal Maps resultantes, identificando puntos en común entre los compañeros y encontrando varias aficiones compartidas, como viajar, hacer deporte, disfrutar de la música y el interés por la tecnología.

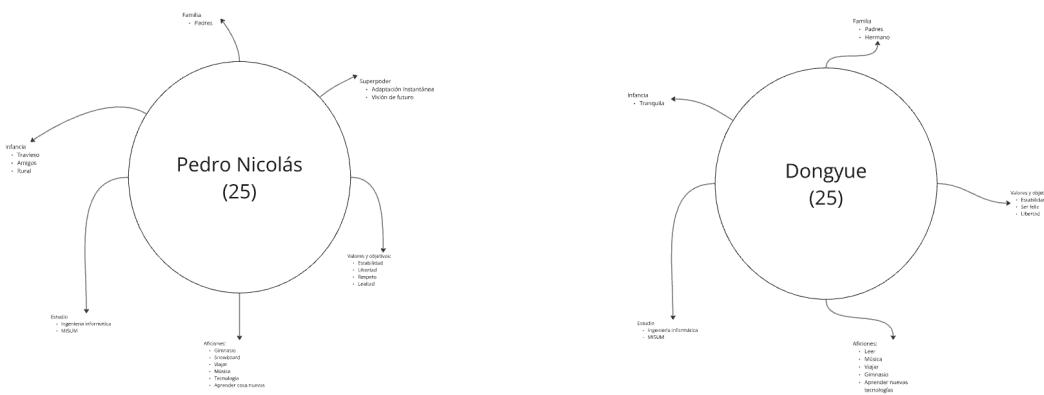


Figura 5. Ejemplos de los Personal Maps del Equipo 1.

La elaboración de los Personal Maps fortaleció la comunicación y el entendimiento mutuo, creando un ambiente más abierto y colaborativo. Al tratarse de un grupo pequeño de solo cinco integrantes, el proceso fue sencillo y fluido, ya que todos tuvimos la oportunidad de participar activamente y compartir tanto aspectos personales como profesionales, lo que consolidó las relaciones dentro del equipo.

2.2.2 Motivado: Moving motivators

La técnica de [Moving Motivators](#), propuesta por Management 3.0, es una herramienta diseñada para reflexionar y comprender las motivaciones intrínsecas de los miembros de un equipo. Basada en los principios de la teoría de la motivación, esta técnica ayuda a identificar qué factores son más relevantes para cada individuo y cómo estos influyen en su desempeño y bienestar en el trabajo.



Para realizar esta tarea, se ha utilizado una tabla estructurada de la siguiente manera: en el eje X se enumeran las motivaciones propuestas por Management 3.0, como Honor, Poder, y Aceptación, entre otras. En el eje Y se incluyen los nombres de los miembros del equipo. Cada integrante colocó 6 post-its, 3 verdes y 3 rojos, para identificar las motivaciones que más le inspiran y aquellas que no lo hacen. Además, en cada post-it se añadieron signos "+" para clasificar las motivaciones según su nivel de importancia.

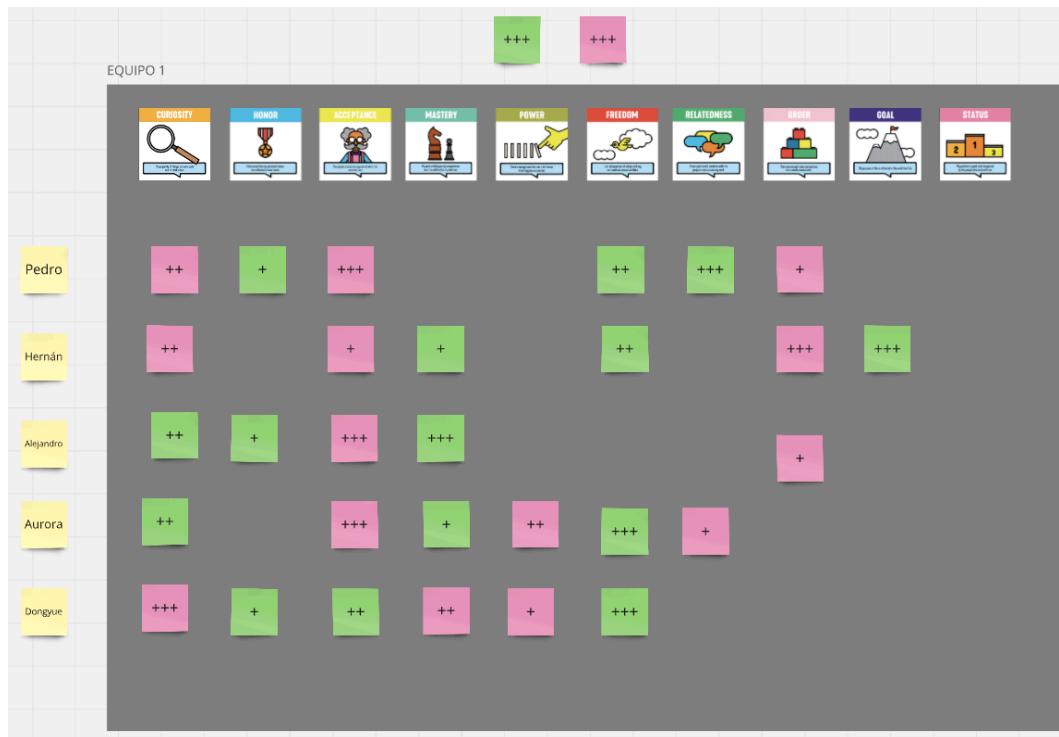


Figura 6. Moving Motivators del Equipo 1.

En la Figura 6, vemos la tabla resultante de los Moving Motivators de nuestro equipo. En ella, se observa una tendencia clara: prácticamente todos los miembros del equipo han identificado las motivaciones de Honor, Maestría y Libertad como importantes para ellos, lo que refuerza los puntos en común y mejora la sinergia dentro del grupo. También se puede observar una tendencia negativa, ya que el equipo está de acuerdo en que la Aceptación y el Orden no son motivaciones que provoquen un mejor rendimiento en el desempeño de su trabajo.

2.2.3 Motivado: Feedback wrap

El [Feedback Wrap](#) es una técnica de retroalimentación propuesta por Management 3.0, utilizada para proporcionar comentarios efectivos en un entorno de trabajo ágil. Su objetivo es mejorar la comunicación, aumentar la motivación y fomentar una cultura de mejora continua dentro de los equipos.

Este enfoque se basa en ofrecer retroalimentación constructiva y específica, enfocándose en comportamientos observables y sus impactos en el equipo o el proyecto. De esta manera, se evita la subjetividad y se garantiza que los mensajes sean claros y accionables.



En nuestro equipo, cada miembro realizó una simulación compartiendo feedback con otro compañero a través del Aula Virtual, con el profesor de la asignatura en copia. Cada mensaje se estructuró según los cinco elementos principales:

1. **Contexto:** Descripción breve del entorno o la situación en la que ocurrió el comportamiento observado.
2. **Observaciones:** Hechos concretos y observables relacionados con la acción o conducta, evitando críticas personales.
3. **Expresa tus emociones:** Haces saber al destinatario como te sientes respecto a los hechos, recalmando el impacto en ti, sin culpar a nadie en particular.
4. **Clasifica según el valor:** Explica por qué los hechos son importantes para ti o para el equipo.
5. **Termina con sugerencias:** Propón acciones concretas para mejorar la situación, dejando espacio para que el receptor proponga otras ideas.

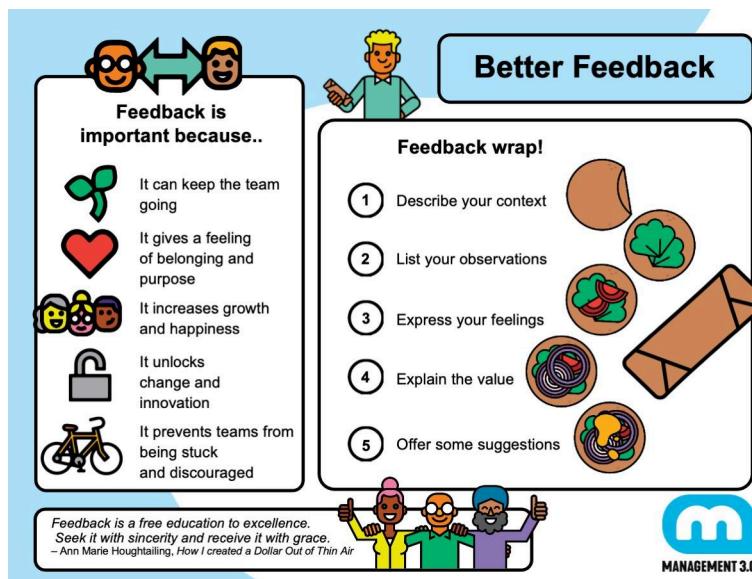


Figura 7. [Ilustración del Feedback Wrap según Management 3.0](#).

A continuación en la Figura 8, se pueden observar los emails enviados/recibidos por los distintos miembros del equipo como parte de la simulación:



Creado por HERVAS LOPEZ, AURORA (aurora.hervas@um.es) (12 nov. 2024 19:57)
Grupo (Estudiante) (7800) GRUPO 1, (7800) GRUPO 1, (7800) GRUPO 1
Para YU, DONGYUE (dongyue.yu@um.es); Destinatario no revelado, (enviada copia a correos de destinatarios)
Asunto [Feedback wrap] Feedback sobre el proyecto SolidarianID
Prioridad Normal

Hola Yue,
Quería transmitirte lo bien estructurada y resuelta que me pareció la implementación del módulo de comunidades que has integrado recientemente al proyecto. Ha sido una aportación importante. Sin embargo, creo que podríamos mejorar la cobertura de tests para facilitar las pruebas y asegurar una mayor confiabilidad.
Gracias por tu dedicación, aprecio trabajar contigo.
¡Saludos!
Aurora



Creado por NICOLAS GOMARIZ, PEDRO (pedro.nicolasg@um.es) (12 nov. 2024 19:59)
Grupo (Estudiante) (7800) GRUPO 1, (7800) GRUPO 1, (7800) GRUPO 1
Para HERVAS LOPEZ, AURORA (aurora.hervasl@um.es); NICOLAS ROS, JOAQUIN (jnr@um.es)
Asunto [Feedback Wrap] Feedback sobre SolidarianID
Prioridad Normal

Hola Aurora,

Quiero compartir un feedback sobre tu trabajo en relación a la implementación del módulo "users" para SolidarianID.

Me ha impresionado mucho cómo has planteado la primera aproximación. El código está bien estructurado y es muy fácil de seguir. Además, los comentarios que añadiste facilitan que el equipo pueda entender y continuar con el trabajo. Esto demuestra un gran nivel de profesionalismo y atención al detalle.

Sin embargo, en algunos casos no has seguido las guías de estilo definidas por la configuración del linter, el cual extiende la configuración base de Airbnb. Esto genera errores en el workflow de GitHub Actions. Sería ideal si pudieras revisar estas peticiones y solucionar los errores que te plantea el comando npm run lint:fix.

A pesar de estos pequeños ajustes, creo que estás haciendo un gran trabajo y aprecio el esfuerzo y la dedicación que estás poniendo en el proyecto. Estoy seguro de que, con estos pequeños ajustes, podrás realizar la integración sin problemas.

¡Gracias por tu trabajo!

Saludos,
Pedro N.



Creado por YU, DONGYUE (dongyue.yu@um.es) (12 nov. 2024 19:54)
Grupo (Estudiante) (7800) GRUPO 1, (7800) GRUPO 1, (7800) GRUPO 1
Para NICOLAS GOMARIZ, PEDRO (pedro.nicolasg@um.es); NICOLAS ROS, JOAQUIN (jnr@um.es)
Asunto [Feedback wrap] Actividad sesión 7
Prioridad Normal

Hola Pedro,

Quería compartirte algunos comentarios sobre nuestro trabajo en el proyecto de solidarianID. Primero, quiero agradecerte por tu excelente colaboración y la disposición que mostraste para trabajar en equipo. La comunicación que mantuviste con el grupo fue clara y constante, lo que realmente ayudó a que todo avanzara sin inconvenientes. Además, tu cumplimiento con los plazos fue clave para que pudieramos mantener el ritmo del proyecto.

Sin embargo, me gustaría sugerir que, en futuras tareas, podamos priorizar de manera más eficiente algunas actividades para evitar retrasos en tareas clave. También creo que podrías participar un poco más en las reuniones del equipo, ya que tu perspectiva es muy valiosa.

En general, fue un placer trabajar contigo. Estoy seguro/a de que con estos pequeños ajustes, tu contribución será aún más efectiva en el futuro. ¡Gracias por todo tu esfuerzo!

Saludos.



ALEJANDRO MONTOYA TORO

Para: @HERNAN RODOLFO SALAMBAY ROLDAN
CC: @JOAQUIN NICOLAS ROS

Hola Hernán,

He aprovechado que tenía un rato libre durante el día de hoy y he revisado tus archivos relacionados con las pruebas del proyecto. He visto que no estás siendo lo suficientemente riguroso con ellas, e incluso algunos archivos .spec.ts se encuentran todavía vacíos.

Me he sentido un poco decepcionado, ya que todo el equipo nos estamos involucrando bastante, invirtiendo tiempo en compaginar la parte de testing con la de desarrollo.

Quiero que comprendas que la parte de testeо de la aplicación es esencial para desarrollar un producto de calidad que nos permita situarnos mejor en el mercado.

Espero que puedas solucionarlo cuanto antes. Si tienes cualquier duda o comentario, no dudes en contactar conmigo.

Muchas gracias por tu atención. Un saludo,



Responder

Responder a todos

Reenviar

| | ...

Mar 12/11/2024 20:08



H HERNAN RODOLFO SALAMBAY ROLDAN
Para: @ ALEJANDRO MONTOYA TORO
CC: @ JOAQUIN NICOLAS ROS

Hola Alejandro,

He estado revisando la documentación de Swagger que has realizado para el módulo usuario y noté que algunos endpoints no tienen descripciones completas en cuanto a los parámetros de entrada, especialmente en los métodos de creación y actualización de usuarios. Esto puede hacer que los desarrolladores nuevos en el proyecto necesiten más tiempo para entender cómo interactuar con la API.

Me sienta un poco mal tener que hacerte apuntes, ya que sé que habías trabajado en las descripciones de otros módulos y habíamos acordado que este también sería detallado de manera similar para asegurar consistencia.

Es importante que esta documentación quede lo más clara y completa posible, especialmente porque nuestro equipo de testing comenzará a usarla pronto. ¿Podrías revisar esos métodos y agregar más detalles sobre los parámetros y respuestas esperadas? También podrías considerar incluir ejemplos de respuestas JSON para los códigos de estado 200 y 400, que ayudarían mucho en la interpretación.

Gracias por tu esfuerzo en este proyecto, y no dudes en decirme si puedo ayudar en algo para que esta tarea se de por completada.

Saludos,
Hernán

Mar 12/11/2024 19:59

Figura 8. FeedBack Wrap de los distintos miembros del Equipo 1.

Este ejercicio nos ha permitido practicar un enfoque estructurado para ofrecer feedback, fortaleciendo la comunicación y fomentando la colaboración. Esta técnica no solo tiene potencial para identificar áreas de mejora, sino que también genera un ambiente de trabajo más empático y orientado al crecimiento.

2.2.4 Auto-organizado: Team barometer

El [Team Barometer](#) es una herramienta visual que permite a los equipos evaluar y monitorear aspectos clave de su dinámica y desempeño de manera rápida y colaborativa. Su objetivo es fomentar conversaciones abiertas y constructivas para identificar fortalezas, áreas de mejora y posibles desafíos dentro del grupo. Para ello, los miembros del equipo califican su percepción en cada área mediante post-its, stickers o marcadores que reflejan diferentes niveles de satisfacción o acuerdo. Esto proporciona una visión clara y compartida del estado actual del equipo, lo que favorece una toma de decisiones informadas para mejorar el ambiente de trabajo y la eficiencia del equipo.

Dentro de este equipo, se ha realizado un diagrama Team Barometer que ha permitido evaluar ciertos aspectos del equipo. Por cada aspecto, cada miembro del equipo ha colocado uno de estos post-it:

- **Post-it rojo:** Indica que el miembro considera que ese aspecto no se cumple o está en un estado crítico dentro del equipo.
- **Post-it amarillo:** Sugiere que el aspecto evaluado está en progreso, pero aún no se percibe como plenamente satisfactorio o que hay margen para mejoras.
- **Post-it verde:** Indica que el miembro considera que ese aspecto está bien desarrollado o que el equipo funciona de manera óptima en esa área.



Figura 9. Team barometer del Equipo 1.

Una fracción del diagrama realizado en la actividad lo podemos ver en la Figura 9. Aquí observamos cómo se están evaluando varias áreas como son la Confianza, Feedback, Compromiso, Responsabilidad Mútua y Orgullo. Cada una de ellas ha sido clasificada de manera positiva por cada miembro del equipo, ya que se han asignado post-it verdes mayoritariamente, lo que indica que los componentes del equipo consideran que esas áreas están desarrolladas y funcionan de manera óptima dentro del grupo. En este diagrama se han evaluado muchos otros aspectos del equipo, no obstante, se ha mostrado sólo una parte de él. El diagrama al completo se puede visualizar dentro de la plataforma Miro.

2.2.5 Auto-organizado: Toma de decisiones: Apocalipsis zombie

El ejercicio del Apocalipsis zombie es una dinámica grupal que pone a prueba la capacidad de auto-organización y toma de decisiones del equipo en una situación hipotética. Este tipo de actividad es común en entornos de trabajo ágiles, ya que permite evaluar y desarrollar habilidades como la colaboración, el pensamiento crítico y la negociación.

Para llevar a cabo la dinámica, en primer lugar se acordó una estrategia de toma de decisiones. Como equipo, optamos por un sistema democrático, lo que facilitó que cada miembro pudiera expresar su opinión y participar activamente en la elección final. Se realizó una votación en la que cada integrante comentó a quién salvaría y tras un debate reflexivo sobre las elecciones individuales, se alcanzó una decisión conjunta, que se refleja en la Figura 10.



Figura 10. Resultado de la actividad *Apocalipsis zombie* del Equipo 1.

El grupo decidió salvar al doctor, por su capacidad para atender emergencias médicas y garantizar la salud del grupo; al piloto de aviones, por su habilidad para manejar vehículos y facilitar la movilidad; a la mujer embarazada y a su marido, para asegurar la continuidad de la especie y proporcionar estabilidad emocional; y a la mujer soltera, para contribuir a la diversidad genética y apoyar la natalidad a largo plazo. Estas elecciones fueron consideradas fundamentales para maximizar las posibilidades de supervivencia del grupo en un contexto extremo.

La actividad permitió reflexionar y debatir sobre las prioridades en un contexto extremo, valorando la importancia de escuchar diferentes puntos de vista. Además, el escenario ficticio del apocalipsis zombie aportó un toque divertido que hizo la experiencia más amena y fortaleció nuestra capacidad de organización y trabajo en equipo.

2.2.6 Auto-organizado: Tableros de delegación

Los [Tableros de Delegación](#) son herramientas visuales diseñadas para fomentar la claridad, transparencia y confianza en los equipos al definir niveles de autonomía y responsabilidad. La idea central detrás de los tableros de delegación es que no todas las decisiones deben ser tomadas de manera unilateral por un líder o completamente delegadas al equipo. En lugar de eso, se define un espectro de siete niveles de delegación, que van desde el control total por parte del líder (nivel 1) hasta la autonomía completa del equipo (nivel 7). Este enfoque promueve conversaciones abiertas sobre quién debe decidir qué y en qué medida, adaptándose a las necesidades y la madurez del equipo.



Figura 11. Tablero de delegación del Equipo 1.

Dentro de la Figura 11 se muestra la configuración del tablero de delegación diseñado para nuestro equipo. Para cada una de las decisiones propuestas, se ha seleccionado un nivel entre los siete disponibles, representado con un post-it amarillo. Este nivel fue acordado mediante un consenso entre todos los miembros del equipo.

Como se observa, en decisiones más relacionadas con el proyecto en sí, como la selección de miembros del equipo, las herramientas a utilizar o la definición de objetivos, la responsabilidad recae más en el manager. Esto se debe a que se consideró que dichas decisiones requieren una mayor supervisión y liderazgo centralizado. Por el contrario, en aspectos como la organización de la jornada laboral o la selección de vacaciones, se otorgó mayor peso a la decisión del equipo, valorando su autonomía y preferencia personal.

2.2.7 Auto-organizado: Matrices de multifuncionalidad

La [matriz de multifuncionalidad](#) es una herramienta que permite visualizar y evaluar las competencias técnicas y habilidades de cada miembro del equipo frente a las necesidades del proyecto. Este ejercicio es fundamental para optimizar la asignación de roles, identificar fortalezas y detectar áreas en las que es necesario fomentar el aprendizaje y la colaboración.

Para nuestro caso de estudio referente al desarrollo de SolidarianID, identificamos las tecnologías y competencias clave necesarias. Estas se organizaron como columnas en la matriz, mientras que en las filas listamos a los miembros del equipo. Cada miembro evaluó su nivel de conocimiento en cada tecnología:

- **★ Estrella:** Especialista en la tecnología o competencia.
- **○ Círculo:** Con nociones básicas o introductorias.

A continuación, en la Figura 12 se presenta la matriz de multifuncionalidad de nuestro equipo:



Equipo 1	Desarrollo de backend	Docker	BBDD	Microservicios	RESTful APIs	Clean Code	Design Patterns	OOP	CI/CD	Testing	Frontend
Aurora	★	★	★	○	★	○		★		○	○
Hernán	★	★	★	○	★	★	★	★	○	★	○
Dongyue	○		★	○		○		★	○		★
Pedro	○	○	★		★	○	○	★		○	○
Alejandro	○	★	○	★	★			★	○	○	

Figura 12. Matriz de Multifuncionalidad Equipo 1.

Al analizar la matriz observamos:

- El equipo posee fortalezas sólidas en áreas como el Desarrollo Orientado a Objetos, Desarrollo de RESTful APIs, Bases de Datos y Docker.
- Existen brechas en áreas como Microservicios, Clean Code, Testing y Frontend, donde algunos miembros tienen nociones básicas, pero se identificaron oportunidades para aprendizaje y colaboración. En particular, en el caso de CI/CD, sería especialmente importante dedicar esfuerzos adicionales a la formación y práctica en esta área para contar con al menos un especialista, ya que un pipeline automatizado y eficiente es esencial para integrar y desplegar los microservicios de manera ágil y confiable.

La matriz de multifuncionalidad nos ha proporcionado una visión clara de las capacidades de nuestro equipo en relación con los requisitos del proyecto. Este ejercicio ha fortalecido la autoorganización del equipo y ha facilitado la planificación estratégica para abordar nuestras áreas de mejora. Además, nos ha permitido asignar roles de manera eficiente, aprovechando las competencias existentes y planificando actividades de formación interna en las áreas con menor experiencia.

2.2.8 Mejora continua – Estrella de mar

La mejora continua es un principio clave en la gestión ágil de proyectos, que se enfoca en el perfeccionamiento constante de los procesos y en la adaptación a los cambios. Dentro de las diversas herramientas utilizadas para promover este principio, una de las más efectivas es la actividad [Estrella de Mar](#). Elegimos esta actividad en lugar de Tablero de Celebración porque creemos que su enfoque proporciona una visión más clara y directa de lo que queremos hacer.

La actividad Estrella de Mar es una herramienta visual que ayuda a evaluar el desempeño del equipo en diversas áreas, con el objetivo de identificar qué aspectos debemos **mantener, mejorar o eliminar**. El proceso se basa en una estrella dividida en varias secciones que representan distintos aspectos del trabajo o procesos del equipo, tales como:

- **Comenzar a hacer:** aquí se identifican las actividades o prácticas del equipo que debe iniciar para mejorar su rendimiento.



- **Más de:** en esta sección, se destacan aquellas acciones que el equipo tiene en cuenta, pero debe hacer con mayor frecuencia.
- **Seguir haciendo:** se enfoca en las prácticas que están funcionando bien y deben mantenerse para continuar con el progreso.
- **Menos de:** se identifican aquellas acciones que el equipo está realizando en exceso, y que sería beneficioso reducir.
- **Dejar de hacer:** en esta parte, se reflexiona sobre los comportamientos que no están aportando valor y que deberían eliminarse.

Para realizar esta actividad, cuyo resultado se puede ver en la Figura 13, el equipo fue reflexionando y aportando su opinión en cuanto a las prácticas de trabajo, aspectos a mejorar y fortalezas que debemos mantener. En general, se destacó la importancia de mejorar la organización mediante un plan de trabajo claro, la documentación del desarrollo y el seguimiento de tareas. Además, se reconoció que las reuniones de control y la colaboración entre los compañeros son puntos fuertes que se deben seguir haciendo.

Por otro lado, se señaló la necesidad de reducir interrupciones y el trabajo de manera aislada, así como dejar de posponer tareas y trabajar sin organización. En conjunto, estas conclusiones nos proporcionan una hoja de ruta clara para seguir avanzando hacia una mejora continua y fortalecer el desempeño del equipo.



Figura 13. Estrella de mar Equipo 1.

2.2.9 Motivado: Calendarios Niko Niko

En este apartado, el equipo lleva a cabo su participación en la elaboración de un [Calendario Niko Niko](#), este calendario es otra de las herramientas propuesta dentro del marco del Management 3.0, diseñada para registrar el estado de ánimo diario de los miembros de un equipo. El principal objetivo de este calendario es proporcionar una visión clara del bienestar emocional del equipo, facilitando la identificación de patrones y posibles áreas de mejora,



contribuyendo a la creación de un ambiente de trabajo saludable y a fomentar la empatía y la colaboración entre los integrantes.

En el calendario elaborado, cada miembro del equipo selecciona diariamente un emoji que representa su estado emocional, utilizando el siguiente sistema de clasificación:

- 😁 **Awesome Day:** Día excepcional, de alta motivación y satisfacción.
- 😊 **Good Day:** Día positivo, aunque no extraordinario.
- 😐 **Not So Good Day:** Día neutral o con ciertos desafíos.
- 😡 **Horrible Day:** Día negativo, asociado a malestar o frustración.

Esta herramienta permite visualizar rápidamente cómo se sienten los integrantes y comprender cómo factores externos e internos influyen en su desempeño y estado de ánimo. Además, facilita identificar los momentos idóneos para planificar tareas, ajustar la carga de trabajo cuando sea necesario, promover actividades que fortalezcan las relaciones y llevar a cabo acciones destinadas a reforzar la cohesión e integridad del equipo.

En nuestro caso concreto, hemos llevado a cabo este seguimiento durante todo el mes de noviembre. Aunque la actividad originalmente estaba programada para realizarse únicamente del 6 al 20 de noviembre, nuestro equipo consideró de interés continuar con la dinámica, por lo que decidimos extenderla y completarla a lo largo de todo el periodo de noviembre. Los resultados obtenidos se observan en la Figura 14:

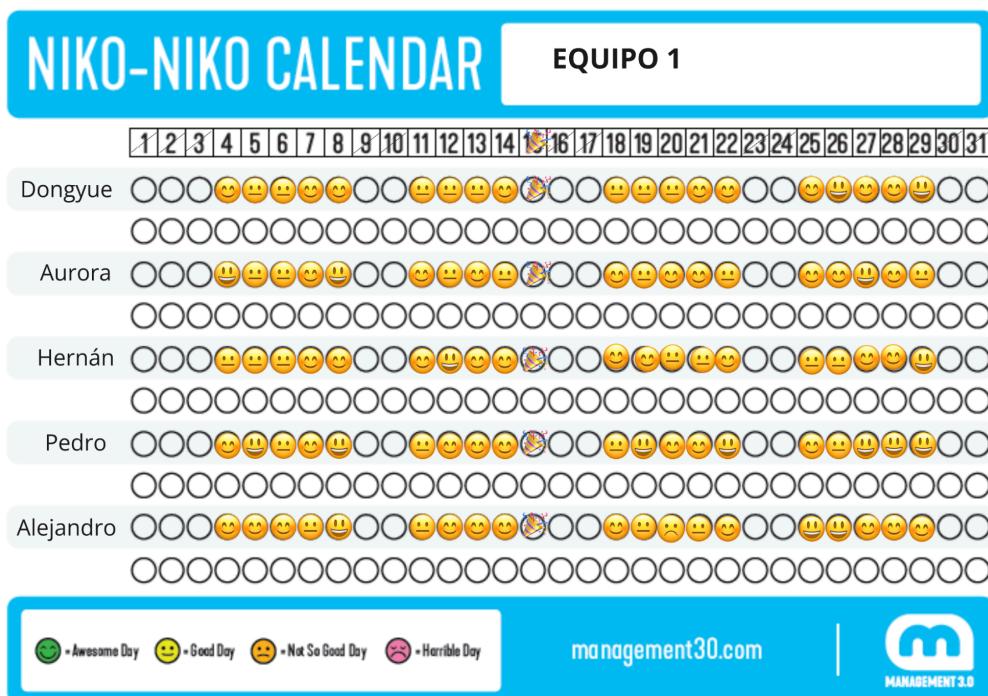


Figura 14. Calendario Niko Niko del Equipo 1 - Noviembre 2024.

Durante el periodo registrado, hemos identificado tendencias que nos han ayudado a reflexionar como equipo. En general, predominan los días positivos (😁 y 😊), lo que nos



confirma que nuestro entorno es motivador y equilibrado. Este resultado refleja un balance adecuado entre nuestras responsabilidades como equipo y nuestro bienestar personal.

También identificamos días puntuales más complejos (😊 y 😞), asociados a tareas exigentes o plazos ajustados que en ocasiones nos llevaron a trabajar en momentos destinados al descanso. Estas situaciones nos llevaron a reflexionar sobre la importancia de ajustar la carga de trabajo y permitir mayor flexibilidad en la realización de tareas, adaptándolas a las necesidades individuales de cada miembro. Asimismo, notamos cómo ciertos eventos colectivos, como las reuniones o entregas, tuvieron un impacto significativo en nuestro estado de ánimo, subrayando la importancia de planificarlos de manera más efectiva para futuras entregas.

Finalmente, el ejercicio nos permitió identificar los momentos idóneos para planificar tareas, fortalecer la cohesión del equipo y celebrar pequeños logros. En conclusión, el **Calendario Niko Niko** ha sido una herramienta útil para monitorear nuestro bienestar, reforzar nuestras dinámicas internas y comprometernos a mantener un entorno de trabajo saludable y productivo.

2.3 Riesgos

En este apartado identificamos, analizamos y proponemos estrategias de mitigación para los riesgos asociados al desarrollo del proyecto SolidarianID, siguiendo la metodología en cuatro etapas propuesta por [Sommerville \(Software Engineering, 10th ed., 2016\)](#). Los riesgos se presentarán clasificados según su naturaleza y priorizados en función de su probabilidad e impacto, con el fin de facilitar su gestión y minimizar su efecto sobre el éxito del proyecto. Los riesgos relacionados con seguridad y privacidad se mencionarán únicamente.

2.3.1 Metodología empleada

La metodología de Sommerville se descompone en cuatro etapas principales:

- **Identificación de Riesgos:** Se enuncian los principales factores internos y externos que podrían afectar al proyecto.
- **Análisis de Riesgos:** Se evalúan probabilidades e impactos de cada riesgo.
- **Planificación de la Gestión de Riesgos (Estrategias de Mitigación):** Se definen acciones preventivas y contingentes para minimizar la probabilidad e impacto de los riesgos más significativos.
- **Monitoreo de Riesgos:** Se establecen mecanismos de seguimiento de los riesgos, permitiendo ajustar las estrategias según evolucione el proyecto.

2.3.2 Identificación y Clasificación de Riesgos

Siguiendo la metodología, iniciamos detallando los riesgos identificados, agrupándolos por categorías:

Riesgos de Proyecto



1. **Cambios frecuentes en los requisitos:** El origen innovador del proyecto y la disponibilidad limitada del Product Owner (PO) pueden ocasionar modificaciones constantes en el alcance y funcionalidades.
2. **Falta de roles clave (QA, UX):** El equipo no cuenta con perfiles especializados en aseguramiento de la calidad (QA) ni en diseño de experiencia de usuario (UX), lo que podría comprometer la calidad y la usabilidad del producto.
3. **Dependencia de verificación manual externa:** La publicación de comunidades, causas y acciones solidarias depende de validaciones manuales por parte de los administradores de la plataforma, lo que puede generar retrasos y cuellos de botella, sobre todo en etapas iniciales si los usuarios administradores no se adaptan con facilidad.
4. **Desviaciones en los plazos y cronograma:** Riesgo de que no se alcancen los objetivos definidos en cada iteración debido a estimaciones incorrectas, bloqueos o impedimentos imprevistos.
5. **Desgaste del equipo y rotación:** Al tratarse de un equipo pequeño, la carga de trabajo acumulada podría derivar en una reducción del rendimiento y en posibles bajas durante el desarrollo.

Riesgos de Producto

6. **Internacionalización incompleta en el MVP:** La falta de soporte multilingüe inicial (idiomas como inglés y español) podría limitar la adopción internacional y afectar a la visibilidad de la plataforma.
7. **Incumplimiento de estándares de accesibilidad:** La falta de experiencia en accesibilidad puede dificultar el cumplimiento de los estándares WCAG 2.2 en su nivel AA, comprometiendo la experiencia de ciertos usuarios.
8. **Falta de feedback temprano de usuarios finales:** Actualmente no existe un proceso formal para validar el producto con usuarios reales, lo que dificulta la identificación temprana de problemas críticos de usabilidad y funcionalidad.
9. **Posposición de funcionalidades avanzadas:** La omisión de características como recomendaciones personalizadas, búsquedas avanzadas y gamificación en el MVP podría reducir la percepción de valor por parte de los usuarios.

Riesgos Técnicos

10. **Escalabilidad y rendimiento:** Aunque nuestra arquitectura propuesta (TS, NestJS, DDD, arquitectura hexagonal) es robusta, no se han definido métricas ni estimaciones de carga, lo que podría derivar en problemas de rendimiento si el sistema crece rápidamente.
11. **Integración con entidades externas (APIs):** No existen datos reales ni acuerdos concretos con ONGs para validar las integraciones mediante APIs, lo que podría retrasar el desarrollo y afectar a la funcionalidad.



- 12. Problemas de infraestructura:** La falta de un entorno adecuado para pruebas o despliegue puede obstaculizar las verificaciones tempranas y la integración continua.

Riesgos de Negocio

- 13. Baja adopción inicial de la plataforma:** No hay una estrategia de marketing clara, lo que nos podría dificultar alcanzar un número adecuado de usuarios y ONGs que hagan funcional la plataforma.
- 14. Dificultades en el cumplimiento normativo (GDPR):** No existe aún un plan definido para asegurar el cumplimiento total de la normativa europea de protección de datos, lo que supone un riesgo de incumplimiento.
- 15. Interés limitado de ONGs y entidades colaboradoras:** Sin acuerdos formales previos, tenemos presente el riesgo de que no se logre el interés suficiente de organizaciones que participen y den visibilidad a la plataforma.

Riesgos de Seguridad y Privacidad (Solo identificación)

- 16. Ciberataques:** Siempre existen riesgos, aunque pequeños, de ataques externos que afecten la disponibilidad e integridad de nuestro sistema. (Por ejemplo: DDoS, inyecciones de código).
- 17. Filtraciones de datos personales:** Riesgo de que existan vulnerabilidades, en nuestro código o en las librerías usadas, que comprometan la privacidad y seguridad de los datos de los usuarios.
- 18. Uso indebido de datos por terceros:** Existe el riesgo de que las entidades externas autorizadas no gestionen correctamente los datos a los que acceden.

2.3.3 Análisis de Riesgos

En cuanto al análisis, la tabla siguiente resume la evaluación de probabilidad y el efecto de cada riesgo:

No	Riesgo	Probabilidad	Efecto
1	Cambios frecuentes en los requisitos	Alta	Serio
2	Falta de roles clave (QA, UX)	Media	Serio
3	Dependencia de verificación manual externa	Media	Tolerable
4	Desviaciones en los plazos y cronograma	Media	Serio
5	Desgaste del equipo y rotación	Baja	Serio
6	Internacionalización incompleta en el MVP	Media	Tolerable
7	Incumplimiento de estándares de accesibilidad	Media	Serio



8	Falta de feedback temprano de usuarios finales	Media	Tolerable
9	Posposición de funcionalidades avanzadas	Baja	Tolerable
10	Escalabilidad y rendimiento	Baja	Serio
11	Integración con entidades externas (APIs)	Media	Serio
12	Problemas de infraestructura	Baja	Tolerable
13	Baja adopción inicial	Media	Serio
14	Dificultades en el cumplimiento normativo	Media	Serio
15	Interés limitado de ONGs	Baja	Tolerable
16-18	Riesgos de seguridad y privacidad	-	-

2.3.4 Planificación de la Gestión de Riesgos

En la siguiente tabla se muestran los riesgos ordenados por prioridad (comenzando por aquellos con mayor necesidad de atención) y una estrategia general para su gestión.

Riesgo	Estrategia
1. Cambios frecuentes en requisitos	Iteraciones cortas, refinamiento continuo y priorización del backlog, y tratar de mejorar la comunicación con el PO
2. Falta de roles clave (QA, UX)	Formación interna, herramientas automáticas de QA, apoyo externo cuando sea necesario y revisiones cruzadas en el equipo
4. Desviaciones en plazos y cronograma	Mejorar estimaciones con métricas históricas, ajustar entregas, renegociar fechas con stakeholders si es necesario
13. Baja adopción inicial	Difusión inicial básica, involucrar ONGs reconocidas, ofrecer incentivos a primeros usuarios, ajustar la propuesta según feedback
7. Incumplimiento de accesibilidad	Recibir capacitación en WCAG 2.2, auditorías, pruebas piloto y mejoras iterativas según feedback
11. Integración con APIs externas	Definir contratos claros, mockear APIs, contactar con ONGs piloto, retrasar integraciones complejas hasta tener acuerdos
14. Cumplimiento normativo (GDPR)	Consultar con expertos legales, auditar y corregir posibles incumplimientos antes del despliegue
5. Desgaste del equipo y rotación	Balancear la carga de trabajo, detectar signos de agotamiento en retrospectivas, ajustar ritmo o posponer entregas no críticas



10. Escalabilidad y rendimiento	Pruebas de carga tempranas, arquitectura escalable, contemplar uso de caché y CDNs, optimizaciones continuas según crecimiento
6. Internacionalización incompleta MVP	Iniciar con 2 idiomas clave y agregar más progresivamente, uso de frameworks de i18n
8. Falta de feedback temprano usuarios	Pruebas piloto internas, encuestas a early adopters, ajustes tras la primera versión
3. Dependencia verificación manual	Estudiar la capacitación o tutorial introductorio a la plataforma
12. Problemas de infraestructura	Entornos de prueba y despliegue adecuados, estudiar una buena estrategia de CI/CD, resolver fallos de forma puntual
15. Interés limitado de ONGs	Contactar con ONGs clave pronto, mostrar prototipos y beneficios para incentivar su participación
9. Posposición de funcionalidades avanzadas	Comunicar las novedades futuras planeadas, enfocarse en el MVP, pero continuar agregando funcionalidad de alto interés en cada iteración
16-18. Seguridad y privacidad	Solo identificación, sin análisis ni estrategias en esta etapa, se estudiarán más adelante

2.3.5 Monitorización de Riesgos

En esta etapa se identifican indicadores para monitorear los riesgos y ajustar las estrategias según sea necesario.

Tipo de riesgo	Posibles indicadores
Proyecto	Incumplimiento de plazos; acumulación de bloqueos; dificultades para cerrar iteraciones
Producto	Elevado número de solicitudes de cambio; quejas de usuarios; incumplimiento de estándares (ej. accesibilidad)
Técnico	Problemas recurrentes de infraestructura; lentitud en el rendimiento; dificultades en la integración con APIs externas
Negocio	Baja adopción de la plataforma; falta de interés de ONGs; retrasos en cumplir requisitos legales (GDPR)
Seguridad y privacidad	-



2.3.6 Impediment Backlog Inicial

En este apartado definimos un Impediment Backlog inicial, el cual incluye los riesgos que supondrían impedimentos actuales que requieren atención inmediata por parte del equipo.

Impedimento	Acción Sugerida
Falta de feedback directo de stakeholders externos	Establecer canales de comunicación más ágiles, reuniones cortas y más frecuentes con PO y ONGs clave
Carencia de perfiles QA/UX dedicados	Formación interna, revisiones cruzadas, apoyo externo puntual si es necesario
Dificultades para definir un plan de cumplimiento del GDPR	Consultar con expertos legales y documentar el manejo de datos siguiendo principios de “Privacy by Design”
Entorno insuficiente para accesibilidad e i18n	Configurar herramientas de auditoría y pruebas tempranas de accesibilidad y multilenguaje
Integración con APIs sin datos reales	Crear mocks, definir contratos de integración, contacto inicial con ONGs piloto

Este análisis de riesgos y el Impediment Backlog inicial nos ayuda a establecer una base sólida para abordar los principales desafíos del proyecto, permitiendo al equipo mitigar riesgos, resolver impedimentos y mantener el desarrollo enfocado en los objetivos clave.

2.4 Escalado

En este apartado realizaremos el escalado ágil del proyecto SolidarianID utilizando [SAFe®](#) (Scaled Agile Framework), un marco de trabajo que ofrece principios y prácticas ágiles para escalar la metodología ágil a nivel organizacional.

El proyecto SolidarianID, promovido por la Unión Europea, tiene como objetivo centralizar la ayuda al tercer mundo y a colectivos desfavorecidos. Requiere una solución tecnológica escalable a nivel continental, adaptada a los idiomas, culturas y necesidades específicas de cada país miembro de la UE. También incluye funcionalidades como la gestión de subvenciones, el diseño de una tarjeta bancaria para donaciones y sistemas de respuesta a emergencias, además de garantizar la accesibilidad desde navegadores web y dispositivos móviles. Para ello, será necesario coordinar equipos internacionales, gestionar múltiples flujos de valor y organizar a los 60 empleados dedicados exclusivamente al proyecto.

Para llevar a cabo el escalado ágil, seleccionaremos la configuración más adecuada del marco SAFe® y definiremos un plan estratégico que permita escalar este proyecto de manera eficiente, maximizando la entrega de valor y cumpliendo con los objetivos establecidos.

El contenido que se presenta en este apartado integra ideas y análisis generado por los cinco miembros del equipo, complementado y estructurado con el apoyo de ChatGPT.



2.4.1 Selección del modelo SAFe®

Los modelos de SAFe® son configuraciones estructuradas que permiten aplicar los principios ágiles a gran escala, adaptándose a las necesidades y complejidad de las organizaciones. Existen varios modelos en SAFe® que incluyen diferentes niveles de coordinación y gestión:

- **Essential:** es el modelo básico que organiza equipos ágiles dentro de un solo grupo de trabajo (ART).
- **Large Solution:** coordina varios grupos de trabajo (ARTs) para gestionar soluciones grandes y complejas.
- **Portfolio:** se enfoca en alinear los proyectos con los objetivos estratégicos de la organización.
- **Full:** combina todos los niveles (Essential, Large Solution, Portfolio) para gestionar toda la organización de manera ágil.

Basando en el enunciado, hemos considerado que la configuración de Large Solution es la más adecuada para el proyecto de SolidarianID debido a las siguientes razones:

- **Escalabilidad Compleja:** el proyecto incluye múltiples subproyectos interrelacionados, como la gestión de subvenciones, desarrollo de aplicaciones multilingües, respuesta a emergencias y lanzamiento de tarjetas bancarias. Esto requiere coordinación en múltiples áreas funcionales y equipos, algo que la configuración elegida maneja eficientemente.
- **Múltiples Equipos y Flujos de Valor:** Con 60 empleados dedicados y la necesidad de coordinar diferentes flujos de valor, esta configuración permite estructurar múltiples Agile Release Trains (ARTs) que trabajen en sincronía.
- **Manejo de Soluciones Grandes:** la configuración elegida incluye prácticas específicas para gestionar soluciones a gran escala, como sería la coordinación entre gobiernos, empresas y ONGs, que es un pilar clave de SolidarianID.

Las razones por no elegir las otras configuraciones son las siguientes:

- **Essencial:** este modelo solo cubre la organización de un solo ART, lo que resulta insuficiente para este proyecto, que involucra varios equipos.
- **Portafolio:** como hemos explicado anteriormente, este modelo se enfoca en la gestión de la estrategia y las inversiones a nivel organizacional pero no aborda específicamente la coordinación de equipos múltiples y la gestión de soluciones grandes que requiere el proyecto de SolidarianID.
- **Full:** aunque cubre los demás niveles, en este caso es demasiado complejo y pesado para lo que necesita el proyecto.

En resumen, Large Solution es el modelo ideal porque está diseñado para gestionar soluciones grandes y coordinar múltiples ARTs, lo que es necesario para un proyecto complejo como SolidarianID, sin la sobrecarga de la gestión estratégica y organizacional que aportan los otros modelos.



2.4.2 Plan de actuación del proyecto

En este apartado se define el plan de actuación del proyecto, que resume cómo se llevará a cabo el proyecto de SolidarianID utilizando la configuración de Large Solution. En nuestro caso hemos elaborado un plan general con cuatro etapas, cada una con su objetivo y las fechas aproximadas de cumplimiento, suponiendo que comenzamos el 1 de enero del 2025.

A continuación, se explicará en detalle cada una de las etapas del plan:

- **Etapa 1: Establecimiento inicial del proyecto (Enero - Marzo 2025)**
 - **Preparación del lanzamiento:** consiste en identificar los flujos de valor clave (operaciones y desarrollo), definir la estructura inicial del Agile Release Train (ART) y los roles claves, y dar formación en SAFe® para todos los equipos involucrados.
 - **PI Planning inicial (1er trimestre 2025):** se planifican los primeros Program Increments (PIs), estableciendo el MVP para SolidarianID. También asegurar la alineación con las necesidades europeas como localización de idiomas, adaptaciones culturales, y funcionalidades críticas.
- **Etapa 2: Desarrollo inicial del MVP (Abril - Septiembre 2025)**
 - **Definición de Capabilities y Features:** se detallan las capacidades principales que debe tener la plataforma y se descomponen en features específicas que serán desarrolladas durante los sprints.
 - **Despliegue incremental:** el desarrollo se hace en iteraciones incrementales mediante una arquitectura de microservicios, lo que garantiza que cada componente del sistema sea independiente, escalable y fácil de actualizar sin afectar al sistema en su conjunto. Además, se realizan pruebas piloto en uno o dos países de la UE para validar la viabilidad técnica y operativa de la solución.
- **Etapa 3: Expansión funcional y geográfica (Octubre 2025 - Junio 2026)**
 - **Iteraciones incrementales:** se amplía el alcance del proyecto incorporando nuevas funcionalidades, como la gestión de emergencias y la administración de subvenciones europeas, que se desarrollarán de manera incremental permitiendo que cada versión de la plataforma se vaya adaptando y mejorando a lo largo del tiempo.
 - **Personalización cultural y lingüística:** se adaptará la plataforma según los requerimientos de cada país miembro de la UE, incorporando traducciones y ajustes culturales según sea necesario, asegurando que la solución sea inclusiva y útil para todos los usuarios.
 - **Despliegue de ARTs adicionales:** se introducen nuevos ARTs especializados en funcionalidades específicas, como la tarjeta bancaria vinculada o la gestión de emergencias.



- **Monitoreo y mejora continua:** se implementan métricas de SAFe® para evaluar el progreso y realizar ajustes según sea necesario incluyendo la medición del rendimiento de los ARTs y el seguimiento de los resultados de las entregas.
- **Etapa 4: Escalado a Configuración de Portafolio (A partir de Julio 2026)**
 - **Introducción de una capa de Portafolio:** debido a que el proyecto crece, se introducirá una capa de portafolio para coordinar todas las soluciones y ARTs.
 - **Gestión de épicas estratégicas:** las épicas son iniciativas de alto nivel que abarcan grandes bloques de trabajo. En esta fase, se gestionan las épicas relacionadas con subproyectos clave, utilizando un Kanban de Portafolio para visualizar, priorizar y hacer seguimiento de estas iniciativas.
 - **Consolidación de métricas globales:** se consolidan las métricas a nivel global para evaluar y reportar el progreso del proyecto a la Comisión Europea.

Tras la explicación del plan de actuación que seguiremos en el proyecto de SolidarianID, habría que destacar los elementos necesarios para que dicho plan sea efectivo. Para ello, tenemos los siguientes elementos:

- **Visión del Producto:** define una visión clara alineada con los objetivos estratégicos de la organización.
- **Program Increments (PIs):** establece ciclos de planificación de 8-12 semanas con objetivos y entregables claros.
- **Backlog de Producto:** priorizar historias y tareas en el backlog de producto y el backlog específico del PI.
- **Agile Release Trains (ARTs):** organizar equipos ágiles (ARTs) para coordinar esfuerzos y cumplir con los objetivos del PI.
- **Capacidades y Features:** definir funcionalidades de alto nivel y desglosarlas en features que pueden ser entregadas en una iteración.
- **Épicas (Epics):** gestionar iniciativas grandes que se desglosan en features, alineadas con la estrategia organizacional.
- **Dependencias:** identificar y gestionar las dependencias entre equipos y ARTs.
- **Recursos y Capacidad:** evaluar la capacidad de los equipos y ARTs para cumplir con los objetivos del PI.
- **Riesgos:** identificar y mitigar riesgos durante la planificación.
- **Metas del PI y Objetivos de los Equipos:** establecer metas claras y alineadas para cada equipo.



- **Eventos de SAFe®:** incluir eventos como PI Planning, revisiones y retrospectivas para asegurar el progreso continuo.

2.4.3 Definición de PI Planning

El propósito del PI Planning es coordinar todos los ARTs, establecer objetivos claros y alineados con la visión estratégica de SolidarianID, y definir un plan detallado para los primeros Program Increments, garantizando que todos los equipos trabajen de manera sincronizada hacia el MVP y los entregables del proyecto.

Dado que el proyecto SolidarianID tiene una alta complejidad y un enfoque SAFe, la planificación de todos los Program Increments para el año 2025 debe abarcar 4 PI de 10 semanas cada uno (5 sprints de 2 semanas). Esto permite iterar sobre el proyecto, realizar entregas incrementales y adaptar la estrategia según las necesidades emergentes.

A continuación se detallará cada uno de los Program Increments planificados:

- **PI1: Enero 1, 2025 - Marzo 15, 2025.** Este PI tiene como objetivo los fundamentos de la plataforma y las primeras entregas claves.
 - **Focus Areas:**
 - Diseño e implementación de la infraestructura básica.
 - Desarrollo del módulo Usuarios (registro, edición, consulta de perfil).
 - Internacionalización inicial (definición de arquitectura multilingüe).
 - Feature clave: Registro y gestión de comunidades.
 - **Entregables:**
 - MVP de la plataforma para pruebas internas.
 - Interfaces funcionales para los módulos de Usuarios y Comunidades.
 - Backend preparado para escalabilidad.
 - Gestión inicial de causas y acciones.
- **PI2: Marzo 17, 2025 - Mayo 24, 2025.** En este PI se centra en la expansión funcional y pruebas pilotos en varios países de la UE.
 - **Focus Areas:**
 - Integración con entidades externas (bancos, ONGs, gobiernos).
 - Módulo de Causas y Acciones listo para pruebas.
 - Soporte completo para idiomas europeos principales.
 - Implementación de seguridad y conformidad legal (GDPR).



- Feature clave: Gestión de eventos solidarios.
- **Entregables:**
 - Prueba piloto en 2 países, por ejemplo España y Francia
 - Versión básica de la aplicación móvil.
 - Funcionalidad básica de gestión de eventos.
 - Reportes iniciales sobre uso y adopción.
- **PI3: Mayo 26, 2025 - Agosto 2, 2025.** El objetivo de esta PI es la optimización y la escalabilidad del proyecto.
 - **Focus Areas:**
 - Optimización de la plataforma basada en feedback de los pilotos.
 - Mejora del rendimiento y la escalabilidad para más usuarios simultáneos.
 - Desarrollo del módulo de Gestión de emergencias (voluntarios, donaciones urgentes).
 - Feature clave: Lanzamiento de la tarjeta bancaria solidaria.
 - **Entregables:**
 - Plataforma operativa en 10 países europeos.
 - Integración funcional con sistemas de emergencia europeos.
 - Tarjeta bancaria funcional y vinculada a la aplicación.
 - Implementación completa del módulo de donaciones.
- **PI4: Agosto 4, 2025 - Octubre 11, 2025.** El objetivo de este PI es la expansión completa del SolidarianID y el cierre de objetivos iniciales.
 - **Focus Areas:**
 - Ampliación a los 27 países de la UE.
 - Mejoras en accesibilidad y soporte de idiomas regionales.
 - Automatización de procesos administrativos.
 - Feature clave: Dashboard de métricas para stakeholders (ONGs, gobiernos, usuarios).
 - **Entregables:**
 - Plataforma completamente operativa a nivel europeo.



- Herramientas analíticas para gobiernos y ONGs.
- Versiones accesibles en navegadores y dispositivos móviles.
- Capacidad para gestionar emergencias y reportes en tiempo real.
- **PI5 (Opcional): Octubre 13, 2025 - Diciembre 20, 2025.** Este PI es opcional ya que tiene como objetivo la innovación y mejoras futuras.
 - Focus Areas:
 - Prototipos para nuevas funcionalidades basadas en feedback.
 - Desarrollo de una API abierta para terceros (empresas y ONGs).
 - Innovaciones tecnológicas (machine learning para optimización de recursos).
 - Feature clave: Soporte avanzado para personalización cultural y religiosa.
 - Entregables:
 - Roadmap para el desarrollo futuro de SolidarianID.
 - Versión estable de la API para terceros.
 - Prototipo de funcionalidades avanzadas.
 - Incrementos en las herramientas de inteligencia artificial y analítica predictiva.

La realización de esta planificación anual nos dará los siguientes beneficios:

- **Flexibilidad:** Cada PI permite ajustar el enfoque según necesidades emergentes.
- **Alineación estratégica:** Todos los incrementos cumplen objetivos específicos de escalabilidad europea.
- **Entregas incrementales:** Cada PI tiene entregables claros que aportan valor tangible.
- **Escalabilidad progresiva:** Los países y usuarios se integran de manera iterativa, asegurando estabilidad.

2.4.4 Arquitectura de ART

En este apartado se detalla la arquitectura de Agile Release Train (ART) que coordina múltiples equipos ágiles. Esta arquitectura de ART deberá cubrir roles clave, equipos y eventos necesarios para gestionar la complejidad del proyecto.



2.4.4.1 Roles claves

Los roles necesarios para gestionar el ART son las siguientes:

- **Release Train Engineer (RTE):** es el responsable de facilitar el ART asegurando que los equipos estén alineados y enfocados en los objetivos del PI.
- **Product Management:** es el encargado de definir y priorizar las Features en arquitectura de ART deberá cubrir roles clave, equipos y eventos necesarios para gestionar la complejidad del proyecto el ART Backlog.
- **System Architect/Engineer:** son los que define la arquitectura técnica de la solución, asegurando que el sistema sea escalable, modular y capaz de integrarse en diferentes países de la UE
- **Business Owners:** son los interesados que ayudan a establecer prioridades y objetivos a nivel del negocio.
- **Product Owner:** son los interesados y los usuarios finales. Prioriza las historias de usuario y las funcionalidades dentro del ART, asegurando que cada equipo entienda y ejecute los requerimientos del Backlog de Producto.
- **Scrum Master:** es el que facilita la coordinación dentro de los equipos y la resolución de impedimentos.
- **Equipo de Desarrollo:** son equipos ágiles responsables de desarrollar las features y capabilities definidas en el Backlog de PI.

2.4.4.2 Composición de los equipos del ART

El ART para SolidarianID estará compuesto por 8-10 equipos ágiles, con un total de 60 personas asignadas al proyecto. Cada equipo será multifuncional y estará enfocado en un área específica:

- **Equipo de Localización y Adaptación:** se encarga de gestionar las adaptaciones de la plataforma a idiomas, culturas y normativas locales. Estará formado por especialistas en localización, traducción y cumplimiento normativo.
- **Equipo de Desarrollo Core:** se encarga de diseñar y mantener las funcionalidades principales de SolidarianID y estará formado por desarrolladores backend y frontend.
- **Equipo de Emergencias:** se encarga de crear herramientas para la gestión de emergencias y está formado por desarrolladores y expertos en integración de APIs geográficas.
- **Equipo de Subvenciones:** se encarga de implementar sistemas de gestión de subvenciones, desde la petición hasta la justificación y estará formado por desarrolladores y analistas de negocio.



- **Equipo de Tarjeta Bancaria:** se encarga de diseñar e integrar la tarjeta bancaria vinculada a SolidarianID y estará formado desarrolladores fintech y expertos en seguridad.
- **Equipo de Mobile Apps:** se encarga de crear aplicaciones móviles accesibles y usables en todas las plataformas y estará formado por desarrolladores iOS y Android, especialistas UX/UI.
- **Equipo de Testing y QA:** se encarga de asegurar la calidad del software, realizar pruebas de integración y validación continua y estará formado por QA testers, ingenieros de automatización.
- **Equipo de DevOps e Infraestructura:** se encarga de gestionar la infraestructura técnica, los pipelines de CI/CD y la arquitectura en la nube y estará formado por Ingenieros DevOps, administradores de sistemas.

2.4.4.3 Eventos del ART

Para garantizar la entrega continua de valor y la coordinación entre equipos, se organizarán los siguientes eventos clave:

- **PI Planning (Planning Increment):** evento que tiene una duración aproximada de 2 días al inicio de cada PI. Se establecen los objetivos de PI, identifica las dependencias entre equipos, asigna Features y planifica iteraciones.
- **Daily Standups:** reuniones diarias rápidas cuyo objetivo es sincronizar los avances, identificar bloqueos y ajustar el trabajo de acuerdo a los objetivos del PI.
- **Iteration Reviews:** evento que tiene lugar al final de cada iteración cuyo objetivo es presentar los incrementos entregados en la iteración a los Business Owners, stakeholders y miembros del ART, y obtener los posibles feedback.
- **Iteration Retrospective:** evento que tiene lugar al final de cada iteración cuyo objetivo es revisar los resultados del PI, identificar mejoras y ajustar procesos.
- **ART Sync:** evento que tiene lugar cada dos semanas, cuyo objetivo es combinar las reuniones de Coach Sync y PO Sync para garantizar la alineación técnica y comercial.
- **System Demo:** evento que tiene lugar al final de cada PI cuyo objetivo es presentar la solución completa desarrollada durante el PI a los stakeholders. Permite validar el progreso y obtener feedback a nivel global.
- **Inspect and Adapt (I&A):** evento que tiene lugar al final de cada PI cuyo objetivo es revisar el rendimiento global del ART y el progreso del proyecto. Se identifican oportunidades de mejora y ajustes necesarios.



2.4.4.3 Herramientas y Tecnologías

Para garantizar la coordinación y eficiencia es fundamental contar con herramientas y tecnologías que faciliten la gestión del trabajo, la documentación y la comunicación entre los equipos. A continuación, se describen las principales herramientas recomendadas para este proyecto y su propósito en cada etapa del desarrollo:

- **Gestión de backlogs:** Jira Align para centralizar las Features y Historias.
- **Colaboración visual:** Miro o MURAL para mapear dependencias y procesos.
- **Automatización de despliegues:** Jenkins o GitLab CI/CD.
- **Seguimiento del progreso:** Tableros Kanban y métricas de SAFe® (como Burn-Up charts).

2.4.5 Identificación de flujos de valor

En el marco del plan para el proyecto SolidarianID, los principales flujos de valor se pueden dividir en dos categorías: operaciones y desarrollo. A continuación se describen brevemente los flujos de valor clave en cada categoría:

- **Flujo de Valor de Operaciones:** son los procesos que entregan valor directamente al cliente o usuario final mediante la ejecución de actividades clave del negocio. En el caso de SolidarianID, incluyen actividades como:
 - **Gestión de Emergencia:** este flujo de valor está enfocado en la respuesta rápida ante situaciones de emergencia. El objetivo principal es asegurar que los países miembros de la UE puedan gestionar eficazmente emergencias y coordinar recursos en tiempo real.
 - **Gestión de Subvenciones Europeas:** este flujo se centra en la gestión de solicitudes y justificaciones de subvenciones europeas para proyectos de desarrollo y apoyo humanitario. El objetivo principal es facilitar y automatizar el proceso de solicitud y distribución de fondos de la UE, asegurando transparencia y eficiencia.
 - **Gestión de Voluntarios:** este flujo de valor involucra la inscripción, gestión y coordinación de voluntarios en el contexto de emergencias y proyectos de desarrollo. El objetivo principal es asegurar que la plataforma facilite el registro, la asignación y el seguimiento de las actividades de los voluntarios.
- **Flujo de Valor de Desarrollo:** son los procesos y actividades necesarios para crear, mantener y mejorar los productos o soluciones tecnológicas. En el caso de SolidarianID, incluye:
 - **Desarrollo de la Aplicación Principal:** este flujo de valor abarca la creación, implementación y mantenimiento de la plataforma digital SolidarianID, incluyendo todas sus funcionalidades principales. El objetivo principal es crear una solución centralizada y escalable para gestionar la información de los usuarios y los recursos destinados a causas específicas en toda la UE.



- **Localización y Adaptación Cultural:** este flujo de valor está enfocado en adaptar la plataforma para diferentes países de la UE, teniendo en cuenta los requisitos lingüísticos, culturales y regulatorios específicos de cada región. El objetivo principal es asegurar que la plataforma sea accesible, comprensible y legalmente conforme en todos los países miembros de la UE.
- **Integración de Sistema Financiero:** este flujo de valor involucra el desarrollo e integración de un sistema financiero que permita a los usuarios realizar donaciones y gestionar fondos para causas específicas. El objetivo principal es implementar una solución financiera segura, eficiente y fácil de usar para realizar y gestionar contribuciones dentro de la plataforma.

2.4.6 Escalado a Configuración de Portafolio

En este apartado escalamos el proyecto de SolidarianID utilizando la configuración de Portafolio de SAFe®. Para ello, definiremos las épicas estratégicas que alineen los objetivos del proyecto con las metas globales.

A continuación, se definirá las épicas con más detalle usando el formato de Epic Hypothesis Statement. Además se usará el Kanban de Portafolio para gestionar dichas épicas.

2.4.6.1 Épicas Estratégicas

Para definir las épicas utilizaremos el formato de Epic Hypothesis Statement, que es una plantilla estructurada en SAFe® para describir una épica. Sirve para detallar la hipótesis de valor detrás de la épica, definiendo qué se pretende lograr, cómo se medirá su éxito y qué impacto tendrá en la organización o el proyecto.

Para el caso del escalado de SolidarianID se definirá dos épicas:

- **Epic 1: Plataforma Europea para la Gestión de Emergencias.**
 - **Hipótesis de la Épica:**
 - **Si desarrollamos** una plataforma integrada para la respuesta a emergencias que permita la localización de desaparecidos, la coordinación de voluntarios y la distribución de recursos en tiempo real,
 - **entonces lograremos** reducir los tiempos de respuesta y mejorar la eficiencia en la gestión de emergencias en toda la UE,
 - **lo cual dará como resultado** un sistema de ayuda más rápido y confiable, aumentando la confianza de los ciudadanos y gobiernos en la plataforma SolidarianID.
 - **Métrica de éxito:**
 - Reducción del tiempo promedio de respuesta en emergencias en un 30%.



- Aprobación de gobiernos de al menos 10 países miembros tras la implementación piloto.
 - Incremento en la cantidad de voluntarios coordinados a través de la plataforma en un 50%.
- **Epic 2: Localización Multicultural y Financiera.**
 - **Hipótesis de la Épica:**
 - **Si implementamos** una solución que permita la localización de la aplicación para idiomas, culturas y normativas específicas de cada país miembro, así como la integración de un sistema de tarjetas bancarias asociadas,
 - **entonces lograremos** una adopción más amplia de la plataforma por parte de gobiernos, ONGs y usuarios individuales en todos los países de la UE,
 - **lo cual dará como resultado** un incremento significativo en el uso de la plataforma y las donaciones realizadas a través de la tarjeta bancaria.
 - **Métricas de éxito:**
 - Aumento del número de usuarios registrados en un 40% tras la localización.
 - Incremento del 20% en el uso de la tarjeta bancaria para donaciones.

2.4.6.2 Gestión de las Épicas con el Kanban de Portafolio

El Kanban de Portafolio es una herramienta en SAFe® utilizada para visualizar, gestionar y priorizar las épicas estratégicas dentro del portafolio. Además, se divide en varias etapas para gestionar las épicas desde su propuesta hasta su implementación y finalización.

A continuación, se describen con detalle las etapas aplicadas a las épicas definidas en el apartado anterior.

- **Funnel** (Propuesta): en esta etapa se identifican las épicas propuestas y se analizan sus beneficios, costos y viabilidad.
 - **Epic 1:** Justificada por la creciente necesidad de una respuesta rápida a emergencias transfronterizas.
 - **Epic 2:** Justificada por la diversidad cultural y lingüística de la UE y la necesidad de un sistema financiero integrado.
- **Analysis** (Análisis): las épicas, una vez aprobada en la etapa anterior para a esta etapa para evaluar su coste, beneficio, riesgo y dependencias. Se crean hipótesis claras y se detallan las capacidades necesarias.



- **Epic 1:** las capacidades incluyen: diseño de un sistema de voluntarios, gestión de recursos y localización de desaparecidos.
- **Epic 2:** las capacidades incluyen: integración de idiomas, normativas locales y sistemas financieros.
- **Portfolio Backlog** (Backlog de Portafolio): Las épicas aprobadas se mueven al backlog de portafolio y se priorizan según su impacto estratégico y retorno de inversión.
 - **Epic 1** es prioritaria por su impacto directo en la gestión de emergencias, una función crítica para la UE.
 - **Epic 2** es prioritaria por su relevancia para la adopción masiva de la plataforma en diferentes países.
- **Implementation** (Implementación): en esta etapa se asignan capabilities y features específicas a los ARTs para su desarrollo.
 - **Epic 1:** se desglosa en features como sistema de registro y gestión de voluntarios y localización de desaparecidos en tiempo real.
 - **Epic 2:** se desglosa en features como módulos de localización lingüística y cultural e integración con entidades financieras para tarjetas bancarias
- **Done** (Completado): en esta etapa se sitúa las épicas que ha sido entregado y que cumple con las métricas de éxitos.
 - **Epic 1:** se mediría la reducción en los tiempos de respuesta en emergencias en los países piloto.
 - **Epic 2:** se analizará el impacto en el registro de usuarios y las transacciones realizadas con la tarjeta bancaria.

2.4.7 Resolución de una Épica

En este apartado resolveremos la épica de Plataforma Europea para la Gestión de Emergencias, donde se crearán Capabilities específicas y se descompondrá en Features (funcionalidades principales) y Enablers (elementos técnicos o de soporte necesarios para habilitar dichas funcionalidades).

2.4.7.1 Creación de los Capabilities

A continuación, se creará los capabilities de la épica de Plataforma Europea para la Gestión de Emergencias:

- **Capability 1: Coordinación de Voluntarios en Emergencias.** Tiene como objetivo crear una solución centralizada para gestionar el registro, la asignación y la comunicación con voluntarios en tiempo real, optimizando recursos para gobiernos y ONGs



- **Capability 2: Sistema de localización de desaparecidos.** El objetivo es integrar reportes y tecnologías de geolocalización para facilitar la búsqueda y la recuperación de personas desaparecidas durante crisis y desastres.

2.4.7.2 Desglose en Features y Enablers

A continuación, se desglosan las capabilities en features y enables:

- Para **Capability 1: Coordinación de Voluntarios en Emergencias**
 - **Features**
 - **Sistema de Registro de Voluntarios:** permite registrar habilidades, ubicación y disponibilidad, asegurando una base de datos confiable.
 - **Canales de Comunicación en Tiempo Real:** coordina voluntarios mediante chat y notificaciones.
 - **Enables**
 - **Integración con Bases de Datos de Ciudadanos:** conectar la plataforma con sistemas existentes de gobiernos y ONGs para validar la identidad y credenciales de los voluntarios.
 - **Creación de APIs de Geolocalización:** optimizan la asignación precisa y rápida de voluntarios.
- Para **Capability 2: Sistema de Localización de Desaparecidos**
 - **Features**
 - **Registro de Reportes de Personas Desaparecidas:** permitir a los usuarios registrar datos sobre personas desaparecidas
 - **Búsqueda de Coincidencias en Bases de Datos:** cruza esta información con datos de hospitales, refugios y otros organismos para localizar desaparecidos rápidamente.
 - **Enables**
 - **Integración con Bases de Datos Externas:** establecer conexiones seguras con sistemas de hospitales, refugios y organismos internacionales para compartir datos relevantes que garanticen la interoperabilidad entre la plataforma y otras instituciones.
 - **Desarrollo de Algoritmos de Búsqueda:** crear algoritmos avanzados de búsqueda y análisis de datos para identificar patrones y coincidencias en los reportes de desaparecidos que mejoren la precisión y velocidad en la búsqueda de personas desaparecidas.



2.5 Seguridad

La seguridad es un aspecto fundamental en el desarrollo de cualquier proyecto, especialmente cuando este maneja información sensible o crítica. En este apartado, se abordan los posibles riesgos de seguridad que afectan al proyecto SolidarianID, tanto desde una perspectiva de gobernanza como técnica, y se proponen medidas para mitigar estos riesgos mediante un enfoque basado en Secure Scrum.

2.5.1 Amenazas SolidarianID, a nivel de gobernanza

Las amenazas a nivel de gobernanza representan riesgos que pueden comprometer la capacidad de una organización para gestionar eficientemente sus recursos, tomar decisiones estratégicas y garantizar el cumplimiento de sus objetivos de forma ética y transparente. Este tipo de amenazas puede afectar la sostenibilidad, la credibilidad y el cumplimiento normativo de un proyecto, lo que resalta la importancia de evaluarlas adecuadamente y establecer medidas para minimizar su impacto.

A continuación, se presentan las principales amenazas a nivel de gobernanza que hemos identificado como factores de riesgo para el éxito del proyecto SolidarianID:

1. **Incumplimiento del RGPD (Reglamento General de Protección de Datos):** Riesgo de manejo inadecuado de datos personales de los usuarios, lo que podría resultar en sanciones legales y pérdida de confianza.
2. **Incumplimiento de Normativas de Accesibilidad:** No cumplir con estándares de accesibilidad web como WCAG 2.2 podría resultar en exclusión de usuarios y posibles sanciones.
3. **Falta de Supervisión en la Creación de Contenido:** Contenido falso o inadecuado publicado por administradores de comunidades podría derivar en problemas legales y daño a la reputación.
4. **Riesgo de Conflictos de Intereses:** Los usuarios podrían utilizar la plataforma para sus intereses personales sobre el bienestar colectivo.
5. **Incompatibilidad de contratos:** Incumplimiento de acuerdos establecidos con ONGs u otras entidades públicas, lo que podría perjudicar las relaciones comerciales y la reputación del proyecto.

Una vez identificadas las amenazas, evaluamos su impacto a través de una herramienta de [Planning Poker Online](#). Tras tres iteraciones, alcanzamos un valor consensuado utilizando una escala del 1 al 10, donde 10 representa el mayor impacto posible. Los resultados obtenidos han sido los siguientes:

Amenaza	Impacto	Justificación
Incumplimiento del RGPD	9/10	La exposición de datos personales puede resultar en sanciones legales severas y pérdida de confianza de los usuarios.



Incumplimiento de Normativas de Accesibilidad	8/10	La falta de cumplimiento con WCAG 2.2 puede excluir a usuarios clave y llevar a problemas legales por discriminación digital.
Falta de Supervisión en la Creación de Contenido	7/10	Contenido inapropiado puede dañar la reputación de la plataforma y generar conflictos legales si no se modera adecuadamente.
Riesgo de Conflictos de Intereses	6/10	Los usuarios podrían actuar por intereses personales, afectando el propósito de la plataforma y creando tensiones internas.
Incompatibilidad de Contratos	8/10	El incumplimiento de contratos puede generar sanciones legales y pérdida de asociaciones estratégicas esenciales.

Tras la evaluación de las amenazas, el siguiente paso fundamental es priorizarlas según su impacto en la organización y proponer posibles medidas de mitigación para minimizar sus efectos en caso de materializarse. Además, sería recomendable definir planes de contingencia que establezcan acciones específicas a seguir si alguna de estas amenazas llegase a ocurrir, aunque este último punto queda fuera del alcance de esta entrega.

Incompatibilidad de Contratos (9/10)
● Realizar auditorías de cumplimiento normativo regularmente.
● Anonimizar datos sensibles y garantizar que se utilicen protocolos de encriptación.
● Designar un rol específico encargado de supervisar estas prácticas.

Incompatibilidad de Contratos (8/10)
● Revisar y renovar contratos periódicamente con el equipo legal.
● Establecer un proceso de resolución de disputas contractuales.

Incumplimiento de Normativas de Accesibilidad (8/10)
● Aplicar estándares de accesibilidad WCAG 2.2 desde la fase de diseño.
● Realizar pruebas regulares de accesibilidad con herramientas especializadas.
● Integrar auditorías externas de accesibilidad que certifiquen el cumplimiento.



Falta de Supervisión en la Creación de Contenido (7/10)

- Establecer un sistema de moderación automática para contenido inapropiado.
- Definir políticas claras de uso y términos de servicio para administradores.

Riesgo de Conflictos de Intereses (6/10)

- Definir un código ético para administradores y usuarios.
- Aplicar sanciones progresivas y medidas disciplinarias en caso de conflictos.

La identificación y priorización de las amenazas a nivel de gobernanza nos ha permitido comprender los riesgos más críticos que podrían comprometer el éxito de SolidarianID. La implementación de las medidas de mitigación propuestas garantiza un enfoque preventivo y proactivo, mejorando la seguridad, la confianza de los usuarios y el cumplimiento normativo. Este análisis fortalece la sostenibilidad del proyecto y asegura que su desarrollo se alinee con los estándares legales, técnicos y éticos necesarios para su éxito a largo plazo.

2.5.2 DFD SolidarianID

En este apartado se muestra y describe el diagrama de flujo de datos ([DFD](#)) para la aplicación SolidarianID. Un DFD es un diagrama que permite modelar el flujo de datos o información dentro de una aplicación o proceso. En concreto, nuestro DFD modela para SolidarianID cómo fluyen los datos desde la parte más externa de la aplicación, es decir, las peticiones realizadas por las entidades (ej: Usuario), hasta la parte más interna de la aplicación, la cual está compuesta por los distintos repositorios (ej: Repositorio de Usuarios).

Este diagrama está compuesto por distintos subdiagramas o niveles, los cuales abordan con más detalle un proceso en concreto del diagrama inmediatamente superior.

Para SolidarianID, hemos realizado un total de 3 diagramas los cuales se describen a continuación:

- **Diagrama de contexto:** acota los límites de la aplicación o proceso, al mostrar el conjunto de entidades con las que se relaciona. Como podemos ver, nuestra aplicación consta de un proceso principal denominado SolidarianID, el cual se relaciona con las entidades Administradores de comunidad, Usuarios y Administradores de plataforma. Cabe mencionar que estas entidades son las únicas consideradas para el Producto Mínimo Viable. Este primer diagrama lo podemos ver en la Figura 15.

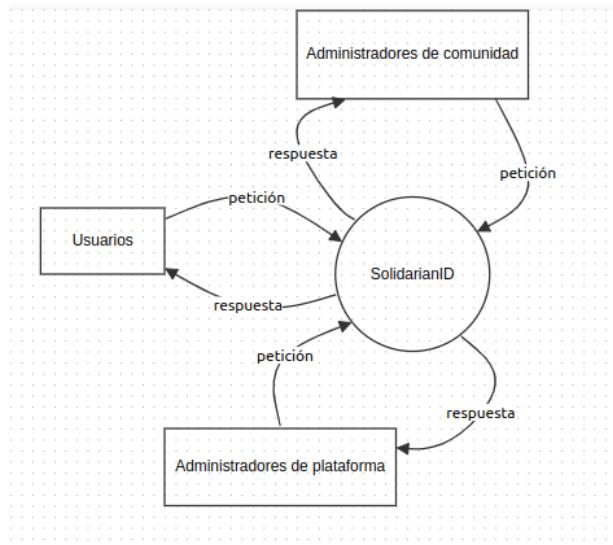


Figura 15. Diagrama del primer nivel.

- **Diagrama de nivel 2:** Aquí se ha realizado la explosión del proceso principal SolidarianID, dando como resultado dos subprocesos que se comunican entre sí: Frontend y Backend. El primero se encargará de la parte gráfica de la aplicación mientras que el segundo contendrá la lógica. Este diagrama lo encontramos en la Figura 16.

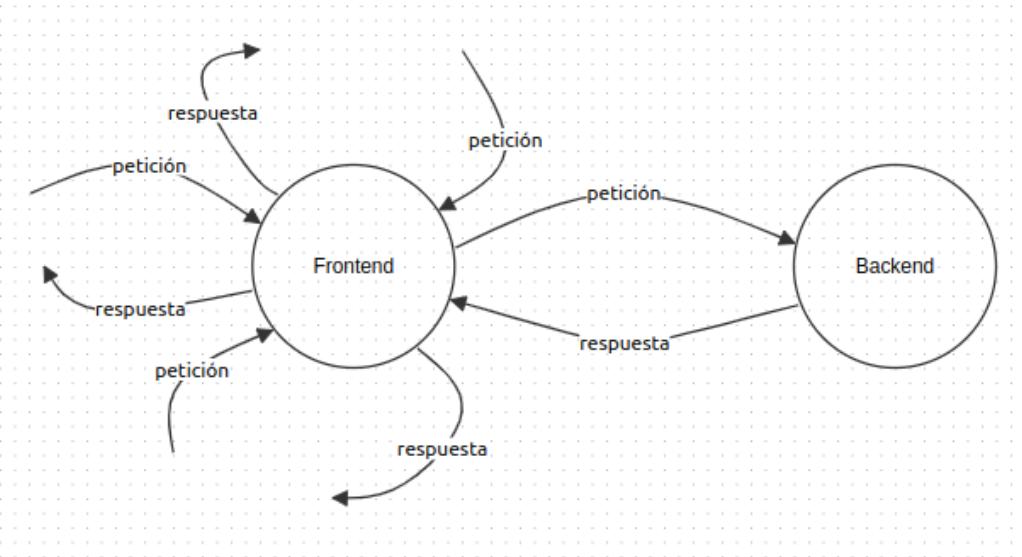


Figura 16. Diagrama del segundo nivel.

- **Diagrama de nivel 3:** En la Figura 17 se ha explotado el proceso Backend generando una serie de subprocesos que corresponden a los distintos microservicios que componen el backend de la aplicación SolidarianID.

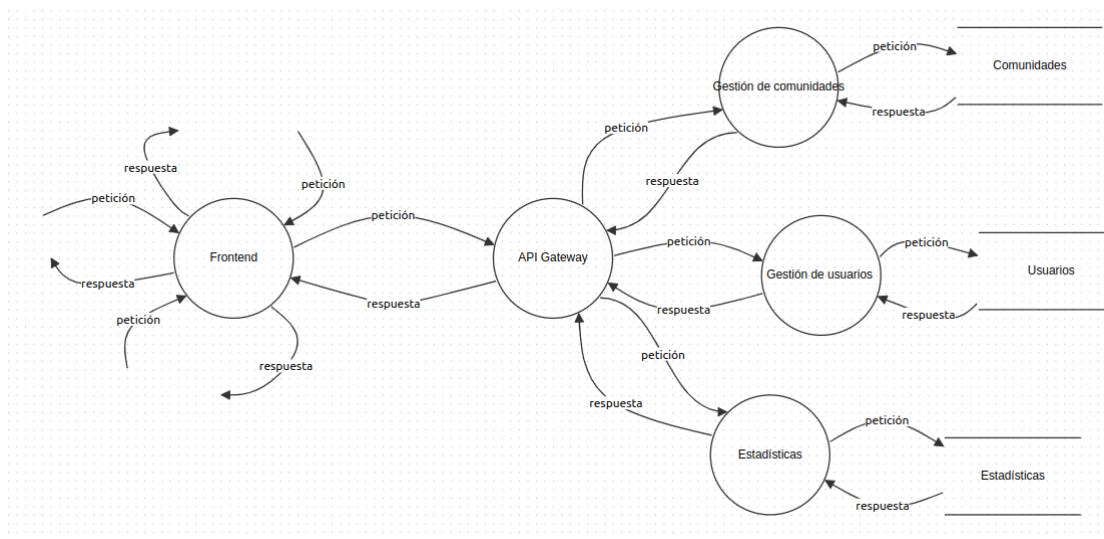


Figura 17. Diagrama del tercer nivel.

Finalmente, se ofrecen una serie de pros y contras sobre la utilidad de estos diagramas en el modelado de amenazas y en la ayuda a la comprensión de los sistemas a construir.

Pros:

- Los DFDs ofrecen una representación visual clara y estructurada de los flujos de información en un sistema, lo que facilita la identificación de posibles vulnerabilidades y amenazas.
- Permiten desglosar el sistema en niveles de detalle, ayudando a comprender mejor los procesos funcionales y cómo interactúan entre sí.
- Sirven como herramienta colaborativa en equipos multidisciplinarios, ya que son fáciles de interpretar incluso para personas no técnicas.

Contras:

- La construcción de DFDs detallados puede ser un proceso laborioso, especialmente en sistemas complejos, lo que puede retrasar los tiempos en proyectos ágiles.
- La técnica no siempre captura todos los aspectos de seguridad necesarios, como controles específicos o dependencias externas, lo que podría requerir complementar con otras herramientas.
- En entornos altamente dinámicos, los diagramas pueden volverse rápidamente obsoletos debido a los cambios constantes en los requisitos o las funcionalidades del sistema.

2.5.3 Amenazas técnicas para SolidarianID

Para abordar la actividad de identificación y análisis de amenazas técnicas del proyecto SolidarianID, hemos seleccionado un diagrama de sistema de microservicios. Este tipo de diagrama proporciona una representación clara y estructurada de la arquitectura del sistema, destacando las interacciones entre componentes clave, como el API Gateway, los microservicios y las bases de datos. Su naturaleza modular facilita la identificación de

superficies de ataque críticas y permite analizar amenazas específicas en cada componente, lo que permite identificar vulnerabilidades y analizar su impacto.

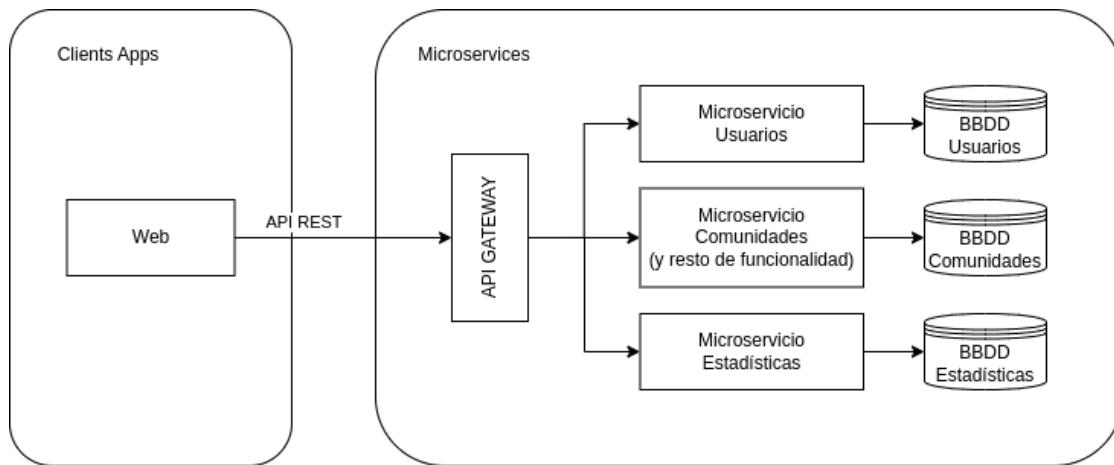


Figura 18. Diagrama de arquitectura de microservicios de SolidarianID.

A través de la Figura 18, se han identificado las superficies de ataque y los puntos críticos de interacción dentro del sistema. Siguiendo el modelo [STRIDE](#), que clasifica las amenazas en Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS) y Elevation of Privilege, se han identificado tres amenazas que podrían comprometer la seguridad del proyecto. Estas amenazas se han relacionado con los elementos específicos del sistema representados en el diagrama. Este enfoque permite llevar a cabo una evaluación sistemática de las vulnerabilidades, facilitando la identificación y priorización de riesgos.

Categoría STRIDE	Amenaza	Descripción	Superficie de Ataque
Spoofing	Uso de tokens JWT expirados.	Un atacante utiliza tokens JWT de los cuales no se comprueba correctamente su validez (revocado, de otro usuario) para acceder.	API Gateway y Microservicio Usuarios.
Tampering	Manipulación de datos en BBDD.	Alteración de información crítica relacionada con causas solidarias o acciones.	BBDD de Comunidades
Denial of Service (DoS)	Sobrecarga de la API REST con solicitudes masivas.	Ataques que saturan la API Gateway e inhabilitan microservicios.	API Gateway

A continuación, evaluaremos cada una de las amenazas identificadas previamente utilizando el modelo [DREAD](#). Este modelo permite valorarlas en función de cinco criterios clave: Damage Potential, Reproducibility, Exploitability, Affected Users y Discoverability. El objetivo es cuantificar el nivel de riesgo asociado a cada amenaza, facilitando así la priorización y la toma de decisiones para mitigar sus impactos potenciales.



Cada criterio será calificado en una escala del 1 al 10, la puntuación total permitirá clasificar las amenazas según su gravedad, estableciendo un marco de referencia para implementar medidas de seguridad efectivas y optimizar la gestión de riesgos en el sistema.

Amenaza	D	R	E	A	D	T	Media	Riesgo
Uso de tokens JWT expirados	8	7	6	9	8	38	7.6	Alto
Manipulación de datos en BBDD	9	6	7	8	7	37	7.4	Alto
Sobrecarga de la API REST	8	8	7	6	7	36	7.2	Medio

Las decisiones tomadas en relación con las amenazas identificadas priorizan aquellas que tienen un mayor impacto en la seguridad y estabilidad del sistema. Se ha dado prioridad al uso de tokens JWT expirados y a la manipulación de datos en bases de datos, ya que estas amenazas comprometen la confidencialidad y la integridad de la plataforma, afectando directamente la confianza de los usuarios de SolidarianID. La implementación de mecanismos de revocación de tokens, cifrado de datos y controles de acceso basados en roles (RBAC) se consideran medidas esenciales para mitigar estos riesgos. Por otro lado, la amenaza de sobrecarga de la API REST ha sido evaluada como crítica para la disponibilidad del sistema, por lo que se recomienda aplicar Rate-Limiting y herramientas de firewall para asegurar un funcionamiento continuo del servicio.

Para concluir, recalcar que el diagrama seleccionado ha sido clave para clasificar y priorizar amenazas, ya que proporciona una visión clara de los componentes críticos del sistema, como el API Gateway, los microservicios y las bases de datos. Esto nos ha facilitado la identificación de superficies de ataque y ha permitido enfocar mitigaciones en los puntos más vulnerables.

2.5.4 Secure Scrum para amenazas técnicas identificadas

Secure Scrum es un enfoque que integra prácticas de seguridad dentro del marco ágil Scrum, asegurando que los aspectos de seguridad sean considerados desde el inicio del desarrollo y a lo largo de todas las fases del ciclo de vida del proyecto. Este modelo combina las actividades y roles habituales de Scrum con prácticas específicas de seguridad, como la identificación de amenazas, inclusión de controles y validación de requisitos de seguridad en cada iteración.

Se ha aplicado el método a las dos amenazas identificadas de mayor riesgo en la actividad anterior de listado de amenazas técnicas para SolidarianID, que corresponde con la primera fase de Identificación de Secure Scrum. Estas amenazas son el uso de tokens JWT expirados y la manipulación de datos en bases de datos.

En primer lugar, la amenaza del uso de tokens JWT expirados (Spoofing), según la clasificación DREAD de la actividad anterior, tiene un nivel de riesgo de 7.6. Esta amenaza se da cuando un atacante utiliza tokens no revocados, pudiendo acceder al sistema de manera fraudulenta. Esta amenaza compromete la confidencialidad y la autenticidad del sistema. Para mitigarla, se podrían implementar mejores prácticas de verificación y validación de tokens.



Por otra parte, la amenaza manipulación de datos en bases de datos (Tampering), tiene un nivel de riesgo de 7.4. Esta amenaza se da por la posibilidad de que un atacante altere información de la base de datos. Afecta a la integridad del sistema y puede dañar la confianza de los usuarios. Como propuesta de mitigación incluimos la implementación de controles de acceso basados en roles para restringir quién puede modificar datos críticos.

Para documentar estos riesgos de seguridad en el Product Backlog y continuar las fases de identificación e implementación, Secure Scrum utiliza las etiquetas **S-tags**. Para implementarlas, se han creado nuevas historias de usuario con su descripción y se han añadido al backlog. En cuanto a las **S-Mark**, se han marcado las S-Tags con la etiqueta *Security* y se han conectado en Jira (con el campo *relates to*) con las historias de usuario relacionadas del MVP, para simular el siguiente diagrama (Figura 19):

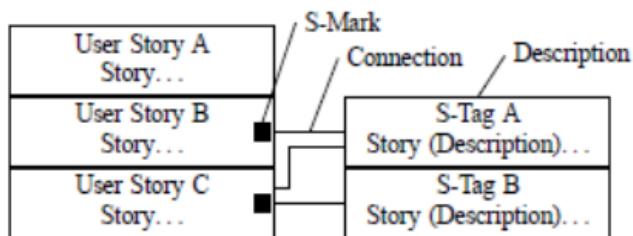


Figura 19. S-Tags y S-Marks en Secure Scrum.

En la Figura 20, se observa cómo quedan las S-Tag añadidas en el product backlog, y posteriormente se proporciona su descripción y criterios de aceptación.

Figura 20. S-Tag de mitigación de amenazas añadidas al Product Backlog.

S-Tag A. Implementación de la validación y verificación de tokens JWT.

Descripción: Como administrador del sistema, quiero que los tokens JWT sean validados y verificados correctamente, para evitar que un atacante reutilice tokens revocados o de otro usuario y acceda indebidamente.

Relacionada con: Login de usuario.

Criterios de Aceptación:

Escenario 1: Revocación de Tokens JWT al cerrar sesión

Dado que un usuario registrado ha iniciado sesión,
cuando pulsa el botón de *cerrar sesión*
entonces el token del usuario debe ser invalidado y añadido a una lista de revocación temporal hasta su expiración.

Escenario 2: Rechazo de Tokens JWT Revocados



Dado que un usuario tiene un token JWT que ha expirado,
cuando realiza una acción en la plataforma
entonces el sistema rechazar la solicitud con un error 401 Unauthorized.

Escenario 3: Reemisión de Tokens nuevos

Dado que un usuario tiene un token revocado o expirado,
cuando realiza un nuevo inicio de sesión,
entonces el sistema debe emitir un nuevo token válido con un tiempo de expiración actualizado y el usuario podrá iniciar sesión.

S-Tag B. Protección de la integridad de los datos críticos en la base de datos.

Descripción: Como administrador del sistema, quiero implementar controles de acceso para garantizar que solo los usuarios autorizados puedan modificar o acceder a datos críticos relacionados con la plataforma, usuarios, comunidades, causas y acciones solidarias.

Relacionada con: Dar de alta una causa solidaria, Crear acción solidaria, Validar creación comunidad, Visualizar informes, Ver estadísticas de uso de la plataforma. (Figura 21)

The screenshot shows a user interface for managing tasks. At the top, there are two items: SOL-212 and SOL-213. Below them is a section titled "Incidencias vinculadas" with a "+" button. Underneath this, the text "relates to" is followed by a list of five items, each with a small icon, a reference number, a task name, a status indicator, and a dropdown menu labeled "TAREAS POR HACER".

Relacionado con	Estado	Tareas
SOL-4 Dar de alta ...	EN PROGRESO	TAREAS POR HACER
Dar de alta una causa solidaria		TAREAS POR HACER
SOL-91 Valida...		TAREAS POR HACER
SOL-98 Visual...		TAREAS POR HACER
SOL-134 Ver e...		TAREAS POR HACER

Figura 21. Historias de usuario relacionadas con la S-Tag de Protección de la integridad de los datos críticos de la base de datos.

Criterios de Aceptación:

Escenario 1: Controles de acceso basados en roles (aplicable a todas las HU relacionadas).

Dado que un usuario con rol de administrador accede al sistema,
cuando intenta leer o modificar datos de las causas o acciones solidarias,
entonces el sistema evalúa el rol y permite la operación.



Dado que un usuario con rol de usuario estándar accede al sistema,
cuando intenta modificar datos de una acción solidaria,
entonces el sistema evalúa el rol y rechaza la operación devolviendo un error 403 Forbidden.

Escenario 2: Prueba de acceso no autorizado (aplicable a todas las HU relacionadas).

Dado que un usuario con rol de usuario estándar accede al sistema,
cuando intenta acceder a un endpoint protegido (/admin/update-cause),
entonces el sistema devuelve un error 403 Forbidden.

Dado que un usuario con rol de administrador accede al sistema,
cuando intenta acceder a un endpoint protegido (/admin/community-report),
entonces el sistema evalúa el rol y muestra el informe de la comunidad.

En cuanto a la verificación de estas S-Tags, se podrían incluir las condiciones necesarias para cumplir una historia de usuario en nuestra Definición de Hecho (DoD):

- *Se ha verificado que los mecanismos de autenticación y autorización implementan controles adecuados para la validez y seguridad de los tokens utilizados.*
- *Se han implementado y validado controles de acceso que garantizan que sólo los usuarios autorizados pueden realizar operaciones sobre los datos críticos del sistema.*

Para concluir, consideramos que el enfoque más adecuado para gestionar los aspectos de seguridad es una combinación de varios métodos como la inclusión de historias de usuario específicas de seguridad en el backlog con criterios de aceptación claros y la definición de requisitos de seguridad en la Definición de Hecho. Este enfoque favorece que la seguridad sea una prioridad integrada en el proceso, fomentando una mejora continua y facilitando la identificación y mitigación de vulnerabilidades a lo largo del desarrollo.

3 Conclusiones

A lo largo de este trabajo, hemos tenido la oportunidad de aplicar distintas metodologías y herramientas que nos han permitido comprender y poner en práctica los conceptos impartidos en la asignatura.

Las actividades relacionadas con la metodología Peopleware fueron las más sencillas de realizar, especialmente aquellas enfocadas en la motivación y cohesión del equipo, como el Personal Map o Moving Motivators. El uso de herramientas como Miro y Jira, con las que ya estábamos familiarizados gracias a su uso durante la asignatura, facilitó la realización de las actividades de manera eficiente.

Por otra parte, algunas actividades relacionadas con el escalado y la seguridad, como la definición del PI Planning dentro del marco SAFe® y el análisis de amenazas técnicas,



resultaron más complejas. Esto se debe a la necesidad de abordar enfoques más estructurados y a un mayor nivel de detalle técnico, especialmente en la identificación y mitigación de riesgos. En este sentido, fue de gran ayuda el uso de herramientas basadas en LLM para la realización de estas actividades.

En cuanto a las actividades más formativas, consideramos que las actividades relacionadas con la autoorganización y la mejora continua, como el Team Barometer y la Estrella de Mar nos brindaron un enfoque práctico para evaluar la dinámica del equipo, identificar áreas de mejora y tomar decisiones de manera colaborativa. Además, la simulación de seguimiento del proyecto con la herramienta Jira resultó especialmente útil, ya que nos permitió gestionar el progreso de SolidarianID de forma visual y práctica, acercándonos a un entorno profesional real. Aunque todas las actividades aportaron valor, algunas como el ejercicio de Apocalipsis Zombie, a pesar de ser creativa y entretenida, tuvo menor impacto en la comprensión técnica de los conceptos.

En conclusión, este trabajo ha sido una experiencia formativa que nos permitió comprender la importancia de la motivación y auto-organización en equipos. Nos ha proporcionado una base sólida para continuar con el desarrollo del proyecto y considerar otras áreas importantes como la gestión de riesgos y la seguridad.

4 Bibliografía

- [Better Feedback Module - Management 3.0.](#)
- [Feedback Wrap -Management 3.0.](#)
- [Equipos Ágiles: Matrices de Multifuncionalidad - YouTube Javier Garzás.](#)
- [Niko-Niko Calendar - Management 3.0.](#)
- [Moving Motivators - Management 3.0.](#)
- [Personal Maps - Management 3.0.](#)
- [Mejora continua: Estrella del mar](#)
- [Team Barmeter - LinkedIn.](#)
- [Tableros de Delegación - Management 3.0.](#)
- [SAFe® - Miro.](#)
- [Agile Release Train - Adaptworks.](#)
- [Modelo STRIDE - Microsoft.](#)
- [El enfoque DREAD para la evaluación de amenazas - Microsoft.](#)
- [Sommerville - Software Engineering](#)
- [DFD](#)