

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**UN GESTOR DE TAREAS BASADO EN KANBAN PARA LA
PLANIFICACIÓN OPERATIVA: UN ENFOQUE ÁGIL PARA LA
GESTIÓN Y ENTREGA CONTINUA DE UN PRODUCTO
SOFTWARE**

**HERRAMIENTAS PARA LA GESTIÓN DEL DESARROLLO ÁGIL
DE UNA APLICACIÓN WEB CON EL MARCO SCRUM**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SOFTWARE**

CHRISTIAN PATRICIO PAZMIÑO BARAHONA

christian.pazmino01@epn.edu.ec

DIRECTOR: JULIO CESAR SANDOBALÍN GUAMÁN

julio.sandobalin@epn.edu.ec

DMQ, julio 2024

CERTIFICACIONES

Yo, Christian Patricio Pazmiño Barahona declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

CHRISTIAN PATRICIO PAZMIÑO BARAHONA

Certifico que el presente trabajo de integración curricular fue desarrollado por Christian Patricio Pazmiño Barahona, bajo mi supervisión.

JULIO CESAR SANDOBALIN GUAMAN
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CHRISTIAN PATRICIO PAZMIÑO BARAHONA

JULIO CESAR SANDOBALIN GUAMAN

DEDICATORIA

Dedico este trabajo a mi madre, por su inquebrantable apoyo y amor incondicional. A mi padre, por su guía y soporte. A mi hermano, por su constante ánimo. A mis profesores y mentores, cuyas enseñanzas y orientación han sido fundamentales en mi formación académica y profesional. A mis amigos, por su constante aliento y compañerismo.

AGRADECIMIENTO

Agradezco a Dios por la salud. A mis padres por su sacrificio. A mi director de trabajo de integración curricular y a mis compañeros de equipo por su apoyo y compromiso.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VI
ABSTRACT.....	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	3
2 METODOLOGÍA.....	18
2.1 Comparación de Herramientas	18
2.2 Configuración de Azure DevOps.....	20
2.3 Planificación del Producto.....	23
2.4 Planificación del Release	25
2.5 Planificación del Sprint.....	30
2.6 Desarrollo de Sprints	31
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	74
3.1 Resultados.....	74
3.2 Conclusiones	79
3.3 Recomendaciones	80
4 REFERENCIAS BIBLIOGRÁFICAS.....	81
5 ANEXOS	83
5.1 ANEXO I	83
5.2 ANEXO II	83

RESUMEN

El presente documento describe el proceso de construcción de un gestor de tareas basado en tableros Kanban para la planificación operativa en organizaciones públicas. Se utilizó el marco de trabajo Scrum y herramientas como Azure DevOps, Microsoft Teams y Miro para gestionar y facilitar el proceso ágil de desarrollo de software. El gestor de tareas se distingue de un producto software tradicional por ser una aplicación adaptativa y compleja.

El desarrollo del componente comenzó con un análisis comparativo de las herramientas disponibles en el mercado, identificando aquellas que mejor se adaptaban a las necesidades del equipo, como la colaboración, el soporte del marco de trabajo y la visualización del progreso del proyecto. Azure DevOps fue elegido por su alta capacidad de integración y soporte para la gestión de tareas y sprints.

Para gestionar el proceso de construcción del producto software, se adoptó el marco Scrum. A lo largo de catorce sprints, el equipo implementó y refinó diversas funcionalidades del gestor de tareas, centradas en la entrega continua de incrementos funcionales que generen valor para el usuario final. El proceso iterativo y adaptativo de Scrum permitió al equipo enfrentar y resolver diversos desafíos durante el desarrollo, como la falta de comunicación y la necesidad de definir claramente la "Definición de Terminado". Cada sprint incluyó sesiones de Daily Scrum, Sprint Review y Sprint Retrospective, promoviendo la transparencia, inspección y adaptación continua.

Al final del proyecto, se logró liberar una segunda versión funcional del gestor de tareas Optiplan, validando así los requisitos y necesidades de los usuarios y mejorando la gestión operativa en el contexto de organizaciones públicas. Las conclusiones indican que el enfoque ágil facilitó la adaptación a los cambios y la entrega de un producto alineado con las expectativas del usuario.

PALABRAS CLAVE: Scrum, Agilismo, Azure DevOps, Desarrollo de Software, Kanban, Gestión Operativa.

ABSTRACT

This document describes the development of a task manager based on Kanban for operational planning in public organizations. The Scrum framework and various tools such as Azure DevOps and Microsoft Teams were utilized to manage and facilitate the agile development process. The task manager distinguishes itself from traditional software products by being an adaptive and complex application designed to meet changing operational needs.

The development of the component began with a comparative analysis of available tools in the market, identifying those that best fit the team's needs, such as collaboration, framework support, and project progress visualization. Azure DevOps was chosen for its high integration capability and support for task and sprint management.

To manage the software product construction process, the Scrum framework was adopted. Over fourteen sprints, the team implemented and refined various features of the task manager, focusing on the continuous delivery of functional increments that add value to the end user. The iterative and adaptive process of Scrum allowed the team to face and resolve various challenges during development, such as communication issues and the need to clearly define the "Definition of Done." Each sprint included Daily Scrum sessions, Sprint Reviews, and Sprint Retrospectives, promoting transparency, inspection, and continuous adaptation.

At the end of the project, a functional first version of the Optiplan task manager was released, validating the requirements and needs of the users and improving operational management in the context of public organizations. The conclusions indicate that the agile approach facilitated adaptation to changes and the delivery of a product aligned with user expectations.

KEYWORDS: Scrum, Agile, Azure DevOps, Software Development, Kanban, Operational Management.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el ámbito de las organizaciones públicas, la planificación operativa es crucial para alcanzar sus objetivos y optimizar los recursos. En este contexto, surge la necesidad de un Gestor de Tareas basado en Kanban, una herramienta diseñada para facilitar la gestión de tareas mediante un enfoque visual y colaborativo.

El Gestor de Tareas no es un producto software tradicional; se trata de una aplicación adaptativa compleja. Su adaptabilidad radica en su capacidad para evolucionar continuamente durante el proceso de construcción, permitiendo ajustes y mejoras en función de las necesidades emergentes del cliente y los aprendizajes adquiridos. Esta evolución es guiada por retrospectivas y principios empíricos, donde el desarrollo iterativo y la retroalimentación de los interesados desempeñan un papel importante. La complejidad del producto se deriva de la incertidumbre inicial sobre los requisitos, los cuales se clarifican y definen progresivamente mediante la interacción con los usuarios y la validación continua de sus necesidades.

El propósito de este componente se centra en la validación de las necesidades de los interesados y la verificación de la entrega de incrementos de software que generen valor. Por ello, en cada iteración se construye un incremento del producto software que se busca satisfacer los requisitos y asegurar que las funcionalidades entregadas aporten un valor real a los usuarios finales.

Para gestionar el proceso de construcción del producto software, se adoptó el marco Scrum, cuyos eventos, como las reuniones diarias, las retrospectivas y las revisiones de los sprints, fueron fundamentales para soportar el ciclo de desarrollo iterativo e incremental en un contexto ágil.

El desarrollo del componente comenzó con un análisis comparativo de las herramientas disponibles en el mercado, identificando aquellas que mejor se adaptan a las necesidades del equipo, tales como la colaboración, el soporte del marco de trabajo y la visualización del progreso del proyecto. La herramienta seleccionada fue Azure DevOps, que es plataforma de Microsoft que ofrece funcionalidades clave para apoyar el marco Scrum, como la gestión del Product Backlog, la planificación de sprints y la documentación colaborativa.

Durante la planificación del sprint, Azure DevOps permitió crear y gestionar el Product Backlog, facilitando la estimación y asignación de tareas. Para el Scrum Daily, se utilizó Microsoft Teams, con reuniones diarias programadas, apoyada por la aplicación de Miro,

que ofreció una pizarra colaborativa en tiempo real para hacer las reuniones más dinámicas.

Durante las revisiones del sprint, el equipo presentaba a los stakeholders el incremento de software a través de una exposición sencilla, mostrando las nuevas funcionalidades y mejoras realizadas. Esta demostración permitía obtener feedback inmediato, identificar áreas de mejora y ajustar las prioridades del backlog.

Finalmente, en la retrospectiva del sprint, se empleó la sección Retrospective de Azure DevOps. Esta herramienta permitió recopilar y priorizar la retroalimentación sobre lo más relevante del sprint, permitiendo al equipo identificar áreas de mejora.

1.1 Objetivo general

Generar valor al usuario final mediante la entrega continua de incrementos funcionales de software.

1.2 Objetivos específicos

1. Implementar herramientas que soporten los eventos y artefactos del marco de Scrum para mejorar la gestión de la construcción de una aplicación adaptativa y compleja.
2. Verificar que las herramientas implementadas soporten el marco de Scrum en la construcción de un Gestor de Tareas Web Basado en Kanban para la gestión operativa en organizaciones públicas.

1.3 Alcance

El alcance del componente se centró en la gestión de la construcción de un producto software mediante el marco de trabajo Scrum, con un enfoque en la entrega continua de valor al cliente final. Con Sprints de dos semanas, el equipo completó un total de 14 sprints durante el desarrollo. Este período de tiempo permitió una validación continua de requisitos y ajustes frecuentes al producto software que facilitaron la entrega continua de incrementos funcionales.

El desarrollo del componente parte desde la planificación del producto, que incluye la definición de su visión, la elaboración del Product Backlog a nivel de épicas, y la planificación del Product Roadmap. Estas actividades establecieron una dirección para la entrega iterativa de incrementos funcionales.

Siguiendo el marco de Scrum, se asignaron roles clave como el Product Owner y el Scrum Master, seleccionados por su experiencia y conocimiento del negocio. El equipo estuvo

compuesto por cuatro estudiantes que desempeñaron varias responsabilidades relacionadas al desarrollo de sus componentes, sin embargo, todos colaboraron estrechamente para asegurar la integración efectiva de sus trabajos individuales y garantizar el valor del producto final para el cliente.

La gestión de eventos Scrum fue fundamental, abarcando Daily Scrums para mantener una comunicación continua, Sprint Reviews para evaluar los incrementos funcionales, y Sprint Retrospectives para mejorar continuamente el proceso de construcción del producto de software.

Durante todo el proceso, se utilizaron las siguientes herramientas: Azure DevOps para la gestión del Product Backlog y Sprint Backlog, Microsoft Teams y Miro para la inspección de las Dailys Scrum. Gracias a la integración con Microsoft Azure por parte de DevOps se facilitó la gestión de despliegues a producción y la revisión de entregables al finalizar el Sprint. La sección de DevOps Retrospectives permitió plasmar los comentarios y retroalimentación del desenvolvimiento del sprint.

Las actividades fuera del alcance del componente incluyeron todas aquellas relacionadas con el Discovery del producto, como la investigación y análisis preliminar del mercado para identificar oportunidades de mejora, la evaluación de la competencia, los estudios de factibilidad, la planificación estratégica del producto y el prototipado con su respectiva validación inicial. Finalmente, es importante mencionar que el componente centró sus esfuerzos en el Delivery del producto, es decir en la entrega continua de incrementos funcionales que generen valor al usuario final.

1.4 Marco teórico

1.4.1 Enfoques Ágiles

La Real Academia Española (RAE) define el término "ágil" como "que actúa o se desarrolla con rapidez o prontitud" (Real Academia Española, 2024a).

La agilidad se entiende como una cualidad de ágil (Real Academia Española, 2024b).

El "Movimiento Ágil" en la industria del software comenzó con la publicación del Manifiesto para el Desarrollo Ágil de Software en 2001, elaborado por un grupo de expertos y consultores en software (Abrahamsson et al., 2017).

Estos enfoques ágiles para desarrollar software se manifiestan en la publicación titulada como el "Manifiesto para el Desarrollo Ágil de Software" (Kent Beck et al., 2001), que define

un conjunto de valores y principios que constituyen la esencia de la agilidad en el desarrollo de software. Los cuatro valores (Kent Beck et al., 2001) se mencionan a continuación:

1. **Individuos e interacciones** sobre procesos y herramientas:

El primer valor del manifiesto enfatiza la importancia de la comunicación y la calidad de los desarrolladores de software por encima de los procesos formales y las herramientas técnicas.

2. **Software funcionando** sobre documentación extensiva:

El manifiesto propone optimizar el tiempo y centrar los esfuerzos en la entrega de software funcional. La documentación intensiva puede ser una limitación, especialmente cuando los requisitos cambian con frecuencia (Sentrio, 2021).

3. **Colaboración con el cliente** sobre negociación de contratos:

La colaboración continua con el cliente es fundamental para entender y cumplir con sus necesidades reales, más allá de lo que se puede establecer en un contrato (Abrahamsson et al., 2017).

4. **Respuesta ante el cambio** sobre seguir un plan:

A medida que el desarrollo del software progresa, desarrolladores y clientes comprenden mejor el sistema, lo que puede llevar a cambios en los requisitos. El manifiesto ágil prioriza responder a estos cambios sobre seguir un plan estricto, con el objetivo de alcanzar la satisfacción del cliente (Ruiz, 2018).

Por otra parte, los **doce principios** (Kent Beck et al., 2001) del software ágil son los siguientes:

1. Satisfacer al cliente a través de entregas tempranas y continuas de software con valor.
2. Hay que reconocer que los requisitos pueden cambiar a medida que se desarrolla el proyecto y está dispuesto a adaptarse a esas modificaciones.
3. Entregar software funcionando con frecuencia, en intervalos de unas pocas semanas a unos pocos meses. El enfoque Agile se centra en entregas frecuentes y regulares para mantener un flujo de trabajo constante (ESIC Business & Marketing School, 2023).

4. Los responsables de negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante todo el proyecto. La colaboración continua entre todas las partes interesadas garantiza una mejor comprensión de los objetivos y requisitos del proyecto.
5. Construir proyectos en torno a individuos motivados. El agilismo reconoce la importancia de la motivación y la confianza en los equipos de desarrollo para fomentar un ambiente productivo (ESIC Business & Marketing School, 2023).
6. El método más eficiente y efectivo para comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software que funciona es la medida principal de progreso. El agilismo valora la funcionalidad real y la entrega de software en pleno funcionamiento (Uribe & Ayala, 2007) .
8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida. Agile busca evitar la sobrecarga de trabajo y promover un ritmo sostenible para el equipo (ESIC Business & Marketing School, 2023).
9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad. El enfoque en la calidad técnica garantiza que el producto sea sostenible y evolucione eficazmente con el tiempo (ESIC Business & Marketing School, 2023).
10. La simplicidad: el arte de maximizar la cantidad de trabajo no realizado es esencial. Eliminar el trabajo innecesario y centrarse en lo que realmente agrega valor es una prioridad en Agile (Uribe & Ayala, 2007).
11. Los mejores diseños, requisitos y arquitecturas emergen de equipos autoorganizados. Fomentar la autoorganización permite que los equipos tomen decisiones más informadas y ágiles (Braude & Bernstein, 2016).
12. A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo y luego ajusta y perfecciona su comportamiento en consecuencia. La mejora continua es fundamental en el agilismo para identificar áreas de oportunidad.

1.4.2 Cynefin Framework

Desarrollado por Snowden y Boone (Snowden & Boone, 2007), el marco de Cynefin es un enfoque valioso que facilita a los líderes la comprensión y actuación en diversos contextos según la naturaleza del problema que enfrentan, permitiéndoles tomar decisiones más efectivas. Este marco se compone de cinco dominios: simple, complicado, complejo, caótico y desorden. Cada uno de estos dominios requiere un enfoque distinto para la toma de decisiones y la resolución de problemas, lo que permite seleccionar las metodologías más adecuadas para cada situación.

En la Tabla 1 se observa una descripción del contexto de cada dominio del marco Cynefin y su enfoque de respuesta.

Tabla 1. Descripción de los dominios que componen el Cynefin Framework

Dominio	Enfoque	Descripción del contexto
Simple	Detectar, Clasificar, Responder	Patrones Repetitivos y eventos consistentes Relaciones claras de causa y efecto
Complicado	Detectar, Analizar, Responder	Diagnóstico de un experto es necesario Relaciones claras de causa y efecto descubribles, pero no evidentes
Complejo	Experimentar, Detectar, Responder	Más impredecible que predecible, requiere experimentación y adaptabilidad
Caótico	Actuar, Detectar, Responder	Muchas decisiones que tomar y sin tiempo para pensar Sin relaciones claras de causa y efecto
Desorden	El objetivo es salir de este dominio	Se está en el dominio del desorden cuando no se sabe en qué dominio está. Es un lugar peligroso para estar porque no se sabe cómo darle sentido a la situación.

En el marco Cynefin, el dominio complejo se presenta como el contexto más adecuado para la implementación del marco Scrum debido a la naturaleza emergente y no predecible de los problemas que se abordan en este entorno (Rubin, 2012).

En el dominio complejo, las soluciones no son obvias desde el principio y solo pueden discernirse iteración tras iteración. Esta característica se alinea perfectamente con el enfoque iterativo e incremental de Scrum, que permite a los equipos explorar, inspeccionar

y adaptar continuamente sus procesos y soluciones basándose en el feedback recibido durante sprints cortos y retrospectivas frecuentes.

1.4.3 Scrum

Scrum es descrito por Ken Schwaber y Jeff Sutherland, como un “marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos” (Ken & Sutherland, 2020).

Por otro lado, Kenneth Rubin en su libro “*Essential Scrum: Una guía práctica*” caracteriza este marco de trabajo como “un enfoque ágil para desarrollar productos y servicios innovadores” (Rubin, 2012).

En este sentido entendemos Scrum como un marco de trabajo ágil diseñado para optimizar la entrega de valor en el proceso de desarrollo de productos o servicios. Este marco de trabajo ofrece una estructura base para el desarrollo adaptativo permitiendo a los equipos responder efectivamente al cambio y la complejidad inherente a proyectos de software complejos.

Scrum se fundamenta en el empirismo y el pensamiento Lean (Schwaber & Sutherland, 2020). El empirismo sostiene que el conocimiento se adquiere a través de la experiencia y la observación, mientras que el pensamiento Lean se enfoca en eliminar desperdicios y concentrarse en lo esencial.

Scrum se rige por tres pilares fundamentales: transparencia, inspección y adaptación, los cuales fomentan la eficiencia y la adaptabilidad en la gestión de proyectos (Atlassian, n.d.).

Transparencia

La transparencia en Scrum es esencial para que tanto el equipo de trabajo como los interesados puedan ver el progreso en el desarrollo de un producto software. Sin una transparencia adecuada, las decisiones pueden resultar en una reducción del valor y un aumento del riesgo. “La transparencia permite la inspección. La inspección sin transparencia genera engaños y desperdicios” (Schwaber & Sutherland, 2020).

Inspección

En Scrum, es crucial inspeccionar regularmente los artefactos y el progreso hacia los objetivos acordados para identificar posibles variaciones o problemas indeseables. “La inspección permite la adaptación. La inspección sin adaptación se considera inútil” (Schwaber & Sutherland, 2020).

Adaptación

Es fundamental ajustar de inmediato cualquier proceso que se desvíe de los límites aceptables o si el producto no cumple con los estándares esperados, con el fin de minimizar problemas adicionales. “La capacidad de adaptación es crucial y se dificulta si los miembros del equipo no están empoderados o no pueden autogestionarse” (Schwaber & Sutherland, 2020).

Eventos de Scrum

Los eventos en Scrum son reuniones estructuradas que se implementan para crear regularidad y reducir la necesidad de reuniones no previstas en el marco. Idealmente, todos los eventos se realizan en el mismo momento y lugar para disminuir la complejidad (Schwaber & Sutherland, 2020).

Tabla 2. Tabla resumen de los eventos de Scrum

Evento	Descripción	Duración Recomendada
Sprint	Ciclo completo de trabajo en el cual se crea un incremento del producto potencialmente entregable	2-4 semanas
Sprint Planning	Planificación del trabajo a realizar en un sprint, se define un objetivo comercial para y se obtiene un Sprint Backlog	2 horas por semana de sprint
Daily Scrum	Reunión diaria para sincronizar actividades, identificar impedimentos y ajustar el plan de trabajo para el siguiente día	15 minutos
Sprint Review	Revisión del incremento del producto y ajuste del Product Backlog según el feedback recibido de los stakeholders	2-3 horas por sprint
Sprint Retrospective	Evaluación del proceso del marco Scrum, identificando mejoras para implementarlas en el siguiente sprint	1 hora por semana de sprint

Sprint

Son los eventos considerados como el esqueleto del marco de Scrum. Poseen una duración fija que no supera el mes calendario (Schwaber & Sutherland, 2020). Las características principales de un sprint son: poseen un límite de tiempo cuya duración es

constante y corta, poseen un objetivo definido previo a iniciar y deben alcanzar el estado final especificado por la Definición de Terminado del equipo.

La Figura 1 presenta una visión general del marco de Scrum, destacando sus eventos clave que facilitan la gestión ágil de productos y servicios. Se da inicio con el Product Backlog, una lista priorizada de requisitos gestionada por el Product Owner, seguida por el refinamiento continuo del mismo. Durante el Sprint Planning, se define un objetivo comercial para seleccionar los requisitos de usuario que permiten alcanzar dicho objetivo y crear un Sprint Backlog. A lo largo de la ejecución del Sprint, el equipo realiza Daily Scrums para inspeccionar el progreso, ajustar el plan de trabajo y abordar impedimentos en la etapa de adaptación. Al final del sprint, se genera un incremento de producto potencialmente entregable, revisado en el Sprint Review. Finalmente, el equipo reflexiona sobre mejoras en la Sprint Retrospective. Más adelante, se explicará con mayor detalle cada uno de estos eventos. Este ciclo de inspección y adaptación fomenta la transparencia, la colaboración y la mejora continua en el proceso de construcción de productos o servicios.

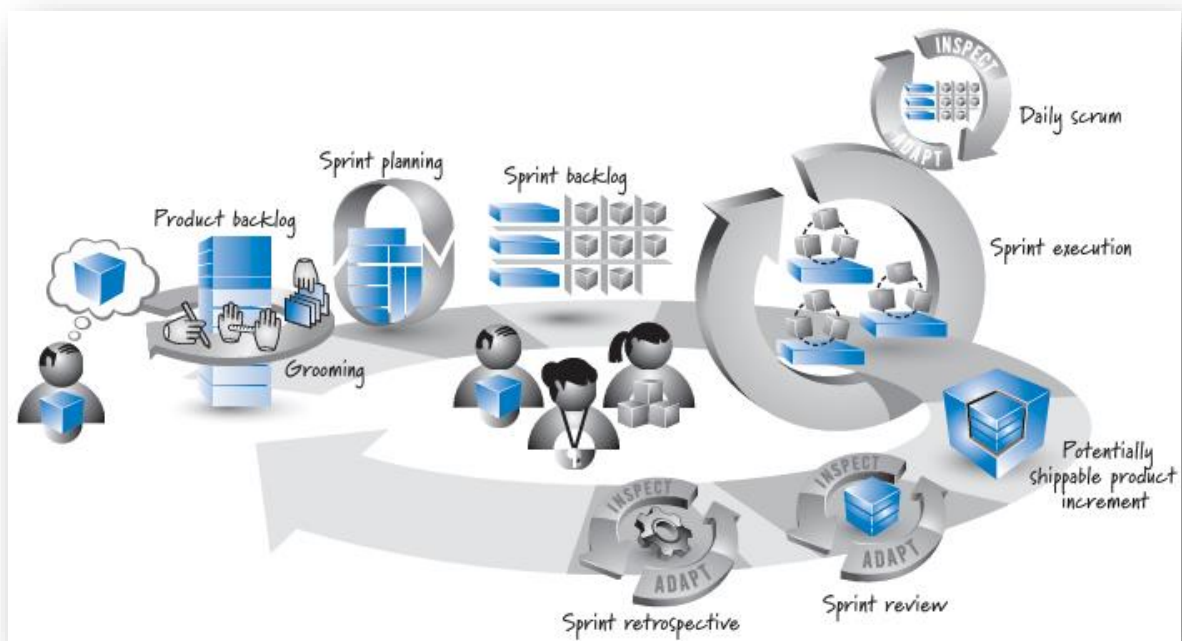


Figura 1. Adaptado de Scrum Framework (p. 17), por K. S. Rubin, 2013, Essential Scrum.(Rubin, 2012)

Sprint Planning

La planificación del Sprint comienza con la especificación del trabajo que se realizará durante el Sprint (Schwaber & Sutherland, 2020). Todo el equipo de Scrum participa en esta tarea. El Product Owner es responsable de preparar a los asistentes para discutir los

aspectos más importantes del Product Backlog y su relación con el objetivo del producto. Además, el equipo Scrum puede invitar a otros interesados para asesorar durante la planificación del Sprint.

La planificación del Sprint aborda tres temas (Schwaber & Sutherland, 2020) principales.

1. ¿Por qué este Sprint es valioso?

El Product Owner guía al equipo sobre cómo mejorar el valor y la utilidad del producto en este Sprint. El equipo de Scrum se organiza para crear un Objetivo de Sprint que explique la importancia del Sprint para las partes interesadas.

2. ¿Qué puede hacerse este Sprint?

Los desarrolladores, junto con el Product Owner, seleccionan elementos del Product Backlog para trabajar durante el Sprint. El equipo de Scrum puede refinar estos elementos para aumentar su comprensión y confianza en ellos. La cantidad de trabajo que se puede hacer en un Sprint depende del conocimiento de los desarrolladores sobre su desempeño previo, su capacidad futura y su Definición de Terminado (Dave West, n.d.)

3. ¿Cómo se llevará a cabo el trabajo seleccionado?

Para cada elemento del Product Backlog, los desarrolladores planifican el trabajo necesario para desarrollar un incremento que cumpla con la Definición de Terminado. Esto generalmente implica descomponer los elementos del Product Backlog en tareas más pequeñas que el equipo de desarrollo puede completar en un día o menos.

El resultado de la planificación del Sprint incluye el Objetivo de la Iteración (Sprint Goal), los elementos de trabajo del producto elegidos para la iteración, conocidos colectivamente como el Sprint Backlog.

Daily Scrum

El objetivo del Daily Scrum es revisar los avances de cada desarrollador en relación con el objetivo del sprint. Estas reuniones, son dirigidas por el Scrum Master y deben ser estructuradas, planificadas a la misma hora y en un máximo de 15 minutos (Schwaber & Sutherland, 2020). Es crucial que todos los miembros del equipo participen y respondan a tres preguntas clave:

1. ¿Cuáles son mis avances desde el último Daily Scrum?
2. ¿Qué puedo lograr antes del próximo Daily Scrum?
3. ¿Qué problemas de bloqueo o impedimentos pueden afectar a mi trabajo?

Durante el Daily Scrum, el equipo inspecciona el progreso hacia el objetivo del sprint, evaluando si las tareas realizadas están alineadas con lo planeado y si el equipo está en buen camino de alcanzar el objetivo del sprint. También se identifican problemas e impedimentos que hayan surgido y que puedan afectar el correcto desenvolvimiento del equipo. En cuanto a la adaptación, el equipo ajusta su plan de trabajo diario para abordar los problemas identificados y optimizar el progreso, lo que puede incluir redistribuir tareas, ajustar prioridades y planificar acciones para eliminar impedimentos.

Sprint Review

El Sprint Review es una reunión al finalizar la ejecución del sprint donde se inspecciona y adapta el incremento del producto software. Durante esta reunión, el equipo Scrum presenta los resultados alcanzados durante el Sprint a las partes interesadas y se discute el progreso hacia el Objetivo de Producto. La inspección se refiere a evaluar el incremento del producto para asegurarse de que cumple con las expectativas del usuario. La adaptación implica identificar cualquier inconveniente o posibilidad de mejora realizar ajustes en el producto software. Es crucial que esta sesión no se limite a una simple presentación, sino que fomente la colaboración activa entre todos los participantes para determinar los siguientes pasos y ajustar el trabajo pendiente del producto según sea necesario (Schwaber & Sutherland, 2020).

Sprint Retrospective

La retrospectiva del Sprint es una reunión crucial en Scrum que se realiza al final de cada ciclo de trabajo. Esta sesión permite al equipo de desarrollo, Scrum Master y Product Owner reflexionar sobre el sprint recién concluido, evaluando qué funcionó bien y qué se puede mejorar. La inspección en esta reunión implica analizar el proceso de trabajo para identificar qué aspectos funcionaron y cuáles no. La adaptación se refiere a identificar e implementar acciones concretas y compromisos para optimizar el proceso de trabajo y las prácticas técnicas del equipo. Estas acciones se implementan en el siguiente sprint para fomentar una mejora continua dentro del marco de Scrum.

Roles de Scrum

Tabla 3. Tabla resumen con las actividades de cada uno de los roles de Scrum

Rol	Actividades
Scrum Master	<ul style="list-style-type: none"> - Facilita el proceso y elimina obstáculos. - Organiza y facilita eventos Scrum. - Coach del equipo en prácticas ágiles y Scrum.

Product Owner	<ul style="list-style-type: none"> - Define la visión del producto y gestiona el Product Backlog. - Prioriza los elementos del Product Backlog según el valor del negocio. - Colabora con stakeholders y asegura que el equipo entienda los requisitos.
Scrum Developers	<ul style="list-style-type: none"> - Desarrolla el producto y entrega incrementos funcionales. - Participa en las reuniones diarias. - Colabora en la estimación y planificación de Sprints.

El equipo Scrum se compone de tres roles fundamentales: el Product Owner, el Scrum Máster y los Desarrolladores.

El **Product Owner** desempeña un papel central como la autoridad responsable de definir y priorizar las funcionalidades del producto, asegurando que el trabajo más valioso se realice en cada iteración. Este rol también implica mantener una visión clara del producto y colaborar activamente con el equipo Scrum para garantizar el éxito general de la solución desarrollada.

El **Scrum Máster** actúa como facilitador y coach, guiando al equipo y a la organización en la adopción y práctica de los valores y principios de Scrum. Además de proteger al equipo de interferencias externas, el Scrum Máster promueve la resolución de problemas y la mejora continua dentro del proceso Scrum. Este rol no ejerce control sobre el equipo, sino que lidera apoyando el desarrollo de un entorno donde el equipo puede prosperar y entregar valor de manera consistente (Rubin, 2012).

Los **Desarrolladores** constituyen un equipo multifuncional y autoorganizado que incluye diferentes habilidades y capacidades técnicas. Su responsabilidad principal es diseñar, construir y probar el producto en cada Sprint. Este equipo trabaja en colaboración estrecha, asegurando que todas las habilidades necesarias estén representadas y comprometidas con la entrega de un software de alta calidad.

Estos roles dentro del equipo Scrum operan en conjunto para maximizar el valor del producto y promover una cultura de mejora continua en toda la organización.

Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor tangible dentro del proceso, diseñados para garantizar la transparencia y registrar la información esencial del proceso de Scrum.

Tabla 4. Tabla resumen de los artefactos de Scrum

Artefacto	Descripción
Product Backlog	Una lista ordenada de requisitos del producto. Es la única fuente de trabajo que debe realizar el equipo de desarrollo.
Sprint Backlog	El conjunto de elementos del Product Backlog seleccionados para alcanzar el objetivo del Sprint, así como un plan para construir y entregar el Incremento del producto software.
Incremento	El resultado de un Sprint. Es la suma de todos los elementos del Product Backlog completados durante un Sprint y los incrementos de todos los Sprints anteriores.

El **Product Backlog** es una lista ordenada de trabajo pendiente que mejora continuamente el producto. Los elementos en ella son refinados y detallados durante la actividad de refinamiento del producto, asegurando que estén listos para ser seleccionados en la planificación del Sprint.

El **Sprint Backlog** es un plan visible que guía el trabajo diario de los desarrolladores durante el Sprint. Nace del Sprint Goal y los elementos seleccionados del Product Backlog, proporciona claridad sobre el qué, cómo y por qué del trabajo a realizar. Se actualiza continuamente para reflejar las decisiones y aprendizajes del equipo durante el Sprint.

El **Incremento** es un avance tangible hacia el Objetivo del Producto, verificado y utilizable al cumplir con la Definición de Terminado. Cada Incremento es aditivo a los anteriores, demostrando progreso continuo y permitiendo revisiones frecuentes para garantizar la calidad y el valor entregado. Este artefacto es crucial para la inspección y adaptación en la Revisión de Sprint, respaldando la transparencia y el empirismo en el proceso de desarrollo. De la mano del incremento está relacionada la **Definición de Terminado**, la cual se traduce como un entendimiento compartido del equipo Scrum sobre lo que significa que el trabajo esté completo, asegurando que los incrementos sean utilizables y cumplan con los estándares de calidad.

1.4.4 Azure DevOps

Azure DevOps promueve un entorno colaborativo y un conjunto de prácticas que unen a desarrolladores, gestores de proyectos y otros colaboradores en el proceso de desarrollo de software. Esta herramienta permite a las organizaciones desarrollar y mejorar sus

productos más rápidamente que con los métodos tradicionales de desarrollo de software (*¿Qué Es Azure DevOps? - Azure DevOps | Microsoft Learn, 2024*). Existen dos modalidades de uso: en la nube mediante Azure DevOps Services, o de forma local con Azure DevOps Server, ofreciendo así flexibilidad según las necesidades de la organización.

Azure DevOps proporciona diferentes procesos para la gestión de elementos de trabajo. La elección adecuada del proceso es fundamental para optimizar el flujo de trabajo y asegurar el éxito del proyecto (*Default Processes and Process Templates - Azure Boards | Microsoft Learn, 2024*). Los procesos predeterminados varían principalmente en los tipos de elementos de trabajo que ofrecen para la planificación y seguimiento del trabajo. Los procesos predeterminados son:

- Básico: Es el más sencillo y está disponible en una vista previa selectiva.
- Scrum: Soporta el marco de Scrum.
- Ágil: Soporta una amplia gama de términos y prácticas ágiles como Scrum.
- CMMI: Brinda el mayor apoyo para procesos formales y la gestión de cambios.

Cuando el equipo utiliza métodos de planificación ágiles, como Scrum, y realiza el seguimiento de las actividades de desarrollo y pruebas de manera independiente, se debe seleccionar el proceso Ágil (*Default Processes and Process Templates - Azure Boards | Microsoft Learn, 2024*). Este proceso es ideal para el seguimiento de historias de usuario y tareas a través de un tablero Kanban ya que proporciona una vista visual del estado de cada elemento de trabajo. Además, las tareas permiten registrar el seguimiento de la estimación original, el trabajo restante y el trabajo completado, lo que ayuda en la planificación y gestión de capacidad del equipo.

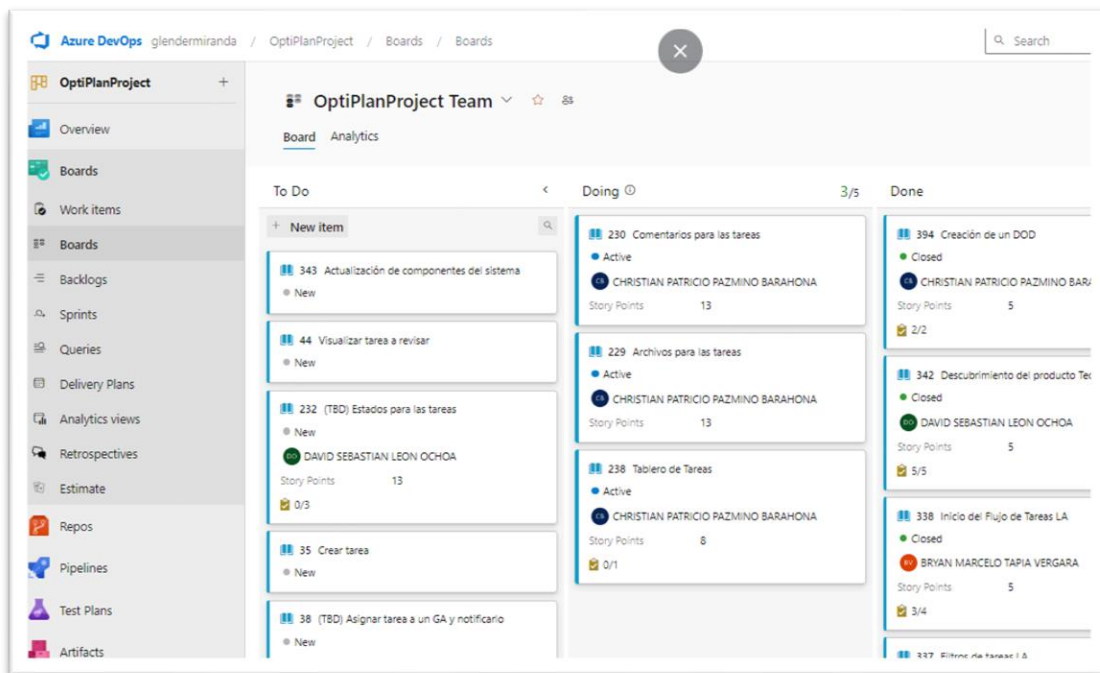


Figura 2. Pantalla general del Board de Azure DevOps

En Azure DevOps, las secciones utilizadas para la gestión de la construcción del producto de software fueron: Overview, Boards y Repos. Estas secciones facilitan la planificación, seguimiento y colaboración en el desarrollo de software, alineándose con los enfoques ágiles como Scrum.

La sección **Overview** proporciona una visión general del proyecto, mostrando información clave sobre el progreso, el estado de las tareas y la salud del proyecto. Dentro de esta sección, la funcionalidad de Wikis es particularmente útil. Las Wikis permiten a los equipos documentar de manera colaborativa información relevante del proyecto, como guías de usuario, especificaciones técnicas y políticas del equipo. Los Wikis son una herramienta vital para mantener a todos los miembros del equipo informados y alineados, facilitando el acceso rápido a la documentación y mejorando la comunicación dentro del proyecto.

La sección **Boards** es una herramienta visual que permite a los equipos gestionar el trabajo mediante tableros Kanban, backlogs y sprints. La página de Boards proporciona una vista general del seguimiento de trabajo mediante paneles Kanban, listas de tareas interactivas y herramientas de planificación. En el Backlog se priorizan las tareas y se planifican las iteraciones (Rea et al., 2024). El Backlog contiene todas las historias de usuario, tareas y bugs que deben ser completados. La página de Sprints ayuda a los equipos a organizar y planificar el trabajo para cada sprint, facilita la asignación de trabajo, el monitoreo del avance diario, y la evaluación de la carga de trabajo, asegurando que los sprints sean realistas y alcanzables (Rohm William Anton et al., 2024).

La sección **Repos** es donde se gestiona el código fuente del proyecto. Utilizando Git se proporciona un control de versiones y herramientas para la colaboración en el desarrollo de software (Microsoft, 2024). Dentro de esta sección está la página de commits, donde cada cambio en el código es registrado, permitiendo un seguimiento detallado del historial de modificaciones del código. La página de branches permiten a los desarrolladores trabajar en nuevas funcionalidades o correcciones de errores de manera aislada antes de fusionar los cambios con el código principal, lo que facilita la integración continua y la entrega frecuente de incrementos de producto (Himanshu, 2024). Finalmente, en la página de Pull Requests los desarrolladores pueden solicitar revisiones de sus cambios antes de fusionarlos, asegurando la calidad y consistencia del código.

1.4.5 Kanban

El significado literal de Kanban es “tarjeta” o “señal” (Castellano Lendínez, 2019).

Kanban es un sistema de gestión del flujo de trabajo en curso, diseñado principalmente para garantizar una producción constante y evitar sobrecargas en el equipo de producción (Bermejo, 2011). En Kanban, la cantidad de trabajo gestionada es exactamente la que el equipo puede manejar. Este sistema de trabajo se basa en el principio "**justo a tiempo**", lo que significa que evita la acumulación de tareas innecesarias, traduciéndose en la gestión de productos software para evitar invertir esfuerzo y tiempo en tareas que no se necesitan de inmediato o que tienen una prioridad más baja, evitando así sobrecargar innecesariamente al equipo.

El sistema Kanban se fundamenta en principios esenciales (Castellano Lendínez, 2019) para su implementación efectiva: visualización completa del flujo de trabajo, enfoque en la calidad desde el inicio, reducción de desperdicios, priorización flexible del tiempo, adaptación continua de las actividades en curso, y mejora constante de los procesos basada en objetivos definidos.

Flujo de Trabajo

El sistema Kanban se centra en la visualización efectiva del flujo de trabajo, utilizando un tablero con tarjetas y columnas. Cada columna representa un paso en el proceso, desde la solicitud inicial hasta la entrega final del producto. Cada tarjeta Kanban representa un elemento de trabajo específico, lo que permite al equipo seguir visualmente su progreso a través del sistema. Esto facilita la identificación de cuellos de botella y áreas de mejora continua, asegurando que el trabajo fluya de manera eficiente y sin interrupciones (Pavel Naydenov, 2022).

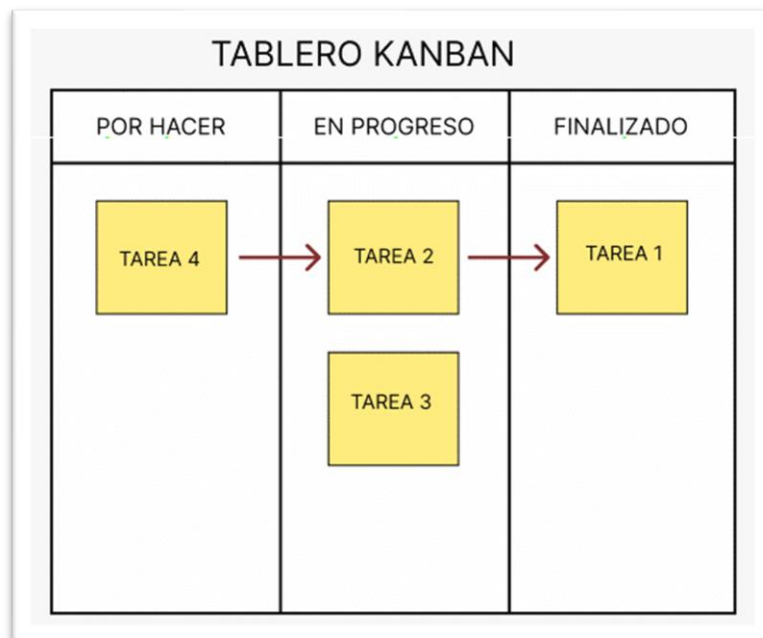


Figura 3. Ejemplo de un Tablero Kanban

En la Figura 3 se observa un tablero Kanban compuesto por tres columnas: "Por Hacer", "En Progreso" y "Finalizado". Las tareas se representan como tarjetas que se mueven entre estas columnas. Las flechas indican el flujo de las tareas a medida que avanzan de una etapa a la siguiente, ilustrando el proceso continuo de trabajo desde el inicio hasta su finalización. Es importante notar que las tareas no necesariamente siguen un orden fijo en este flujo y pueden moverse según las prioridades y el progreso del equipo.

El "Trabajo en proceso" o "**Work in progress**" (por sus siglas en ingles WIP) es el número máximo de tareas activas que un equipo puede manejar simultáneamente en cada etapa del desarrollo. Limitar el WIP es fundamental para optimizar el flujo de trabajo y evitar la sobrecarga del equipo, lo que permite una atención más focalizada en las tareas en curso y facilita la entrega oportuna y de alta calidad de los productos o incrementos del proyecto.

2 METODOLOGÍA

2.1 Comparación de Herramientas

La comparación de herramientas que soportan el marco Scrum es un proceso crucial para seleccionar las más adecuadas que promuevan la colaboración, transparencia y eficiencia del equipo. Herramientas como Azure DevOps, Jira y Asana se destacan por sus funcionalidades específicas que facilitan la gestión del Product Backlog, la planificación de Sprints y la monitorización del progreso. Azure DevOps, por ejemplo, ofrece una integración robusta con otras herramientas de Microsoft que facilitan la gestión de proyectos. Jira, por su parte, es conocida por su flexibilidad y capacidad de personalización, adaptándose a las necesidades específicas del equipo. Asana se destaca por su interfaz intuitiva y su capacidad para gestionar tareas y proyectos de manera visual.

En la Tabla 5 se observa la comparativa de herramientas para la gestión del Product Backlog. Los criterios para la comparativa fueron los siguientes: las capacidades de integración, si el equipo posee experiencia previa en dicha herramienta, el tipo de licenciamiento, el número de usuarios permitidos, la capacidad para gestionar la documentación y el soporte para la colaboración en tiempo real.

Tabla 5. Tabla comparativa de herramientas para la gestión del Product Backlog

Herramientas / Criterios	Azure DevOps	Jira	Asana
Capacidades de Integración	Alta, robusta integración con herramientas de Microsoft	Alta, numerosas integraciones y plugins	Moderada, integraciones a través de API
Experiencia Previa del Equipo	Sí	Sí	No
Colaboración en Tiempo Real	Alta, con Microsoft Teams y Wikis	Alta, con comentarios y notificaciones	Alta, con tareas y comentarios en tiempo real
Gestión de la Documentación	Alta, con Wikis	Moderada, con Confluence	Baja, menos enfoque en documentación
Tipo de Licenciamiento	Licencia Estudiante EPN	Software Libre	Plan Gratuito con limitaciones
Número de Usuarios	5	10	10

Tras evaluar la comparativa para la gestión del Product Backlog en la Tabla 5, se escogió la herramienta de Azure DevOps ya que ofrece una alta capacidad de integración con otras soluciones de Microsoft, el equipo ya tiene experiencia previa utilizándola y como

estudiantes contamos con una licencia estudiantil gratuita proporcionada por la Escuela Politécnica Nacional.

En la Tabla 6 se muestra una comparativa de herramientas para la Daily Scrum en donde se consideraron los siguientes criterios: tipo de licenciamiento, experiencia previa del equipo, capacidades de comunicación en tiempo real (chat, audio o video), acceso multiplataforma (web, móvil, escritorio), límite de duración de llamadas y almacenamiento del historial de reuniones. Las herramientas comparadas fueron Microsoft Teams, Zoom, Google Meet y Slack.

Tabla 6. Tabla comparativa de herramientas para la Daily Scrum

Herramientas / Criterios	Microsoft Teams	Zoom	Google Meet	Slack
Tipo de Licenciamiento	Licencia Estudiantes EPN	Gratuita con Limitaciones	Gratuita	Gratuita con limitaciones
Experiencia Previa	Si	Si	No	No
Capacidades de Comunicación en Tiempo Real	Chat, audio, video	Chat, audio, video	Chat, audio, video	Chat, audio, video
Acceso Multiplataforma	Web, Móvil, Escritorio	Web, Móvil, Escritorio	Web, Móvil	Web, Móvil, Escritorio
Límite de Duración de Llamadas	24 horas (suscripción)	40 minutos (gratis), ilimitado (pago)	60 minutos (gratis), ilimitado (pago)	Ilimitado
Almacenamiento del Historial de Reuniones	Sí (OneDrive, SharePoint)	Sí (local y en la nube)	Sí (Google Drive)	Sí (integraciones)

Como se observa en la Tabla 6 la herramienta que mejor cumple con todos los criterios establecidos es Microsoft Teams. Por tal motivo, se seleccionó dicha herramienta para las reuniones diarias a ejecutarse durante los sprints. Microsoft Teams ofrece una alta capacidad de integración con Office 365, se cuenta con experiencia previa por parte del equipo y proporciona comunicación en tiempo real a través de chat, audio y video.

En la Tabla 7 se realiza una comparativa de herramientas que soporten las Retrospectivas del marco Scrum, en donde se consideraron los siguientes criterios: integración con otras herramientas de gestión de proyectos, capacidad para capturar y gestionar feedback,

generación de reportes y seguimiento de acciones, cantidad de usuarios soportados, disponibilidad de plantillas predefinidas y tipo de licenciamiento. Las herramientas sujetas a la comparación fueron el plugin de Retrospectiva de Azure DevOps, Miro y EasyRetro.

Tabla 7. Tabla comparativa de herramientas para la Retrospectiva

Herramientas / Criterios	Retrospectiva Azure DevOps	Miro	EasyRetro
Tipo de Licenciamiento	Licencia Estudiantes EPN	Gratuita	Gratuita con limitaciones
Integración	Alta, con herramientas de Microsoft	Alta (Trello, Slack, Jira)	Moderada, integraciones limitadas (Jira)
Captura y Gestión de Feedback	Alta, permite registrar y seguir acciones en DevOps	Moderada, con opciones para agregar notas y comentarios	Moderada, captura de feedback básica
Disponibilidad de Plantillas	Moderada, plantillas estándar	Alta, múltiples plantillas	Alta, múltiples plantillas
Cantidad de Usuarios	Ilimitados	Ilimitados	10

Tras evaluar las diferentes herramientas como se ilustra en la Tabla 7 se seleccionó el plugin de Retrospectivas de Azure DevOps. Esta elección se debe a su fuerte integración con otras soluciones de Microsoft, su capacidad efectiva para capturar y gestionar feedback, y su funcionalidad de seguimiento a través de creación de nuevas tareas dentro de DevOps. Además, el plugin soporta un número ilimitado de usuarios y ofrece plantillas estándar que son adecuadas para las necesidades del equipo.

2.2 Configuración de Azure DevOps

Al crear un proyecto, se selecciona un proceso o plantilla de proceso basado en el modelo de proceso de la organización. El término “modelo de proceso” se entiende como “el modelo utilizado para soportar proyectos creados para una organización o colección de proyectos” (chcomley et al., 2024). Por otro lado, el término "proceso" se refiere a la “definición de los componentes básicos del sistema de seguimiento de elementos de trabajo” (chcomley et al., 2024). En base a lo descrito anteriormente se optó por seleccionar el proceso predeterminado “Agile”, el cual permite muchos términos del enfoque ágil, en donde se incluye el marco de trabajo Scrum. Dicho proceso es muy útil cuando se desea dar seguimiento a las historias de usuario en el panel Kanban y a las tareas en el panel de tareas. Algo clave, para el equipo de desarrollo fue que las tareas admiten: el seguimiento de la estimación original, el trabajo restante y el trabajo completado; lo que facilita una

gestión del progreso y la carga de trabajo. En la Figura 4 se observa la jerarquía de elementos de trabajo disponibles para el proceso “Agile”.

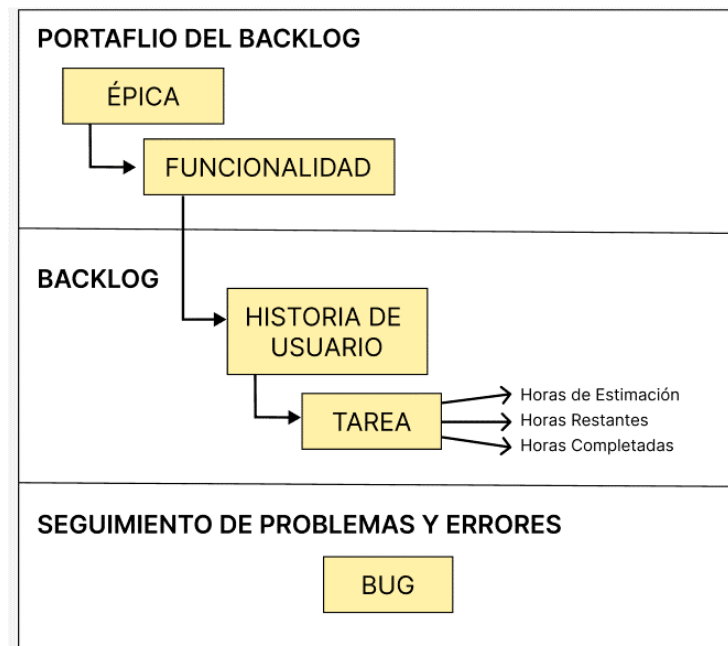


Figura 4. Elementos de Trabajo del Proceso Agile en Azure DevOps

En el nivel superior de la Figura 4 (Portafolio del Backlog), las épicas representan grandes elementos de trabajo que se desglosan en funcionalidades más pequeñas. Si bien los elementos de trabajo llamados “funcionalidades” sí están presentes en el proyecto de Azure DevOps, únicamente se utilizaron como una forma de organización para el equipo y no tuvieron un enfoque significativo dentro del marco Scrum. En el backlog, las historias de usuario detallan características desde la perspectiva del usuario final, desglosadas en tareas específicas que incluyen horas de estimación, horas restantes y horas de trabajo completadas. También, en el proceso Agile están presentes los bugs, que son elementos de trabajo dedicados al registro y seguimiento de errores. Esta estructura jerárquica facilita la organización, priorización y seguimiento del progreso de la construcción del producto de software.

Una configuración importante en Azure DevOps fue la habilitación del plugin "Retrospectives". Esta extensión permitió al equipo realizar retrospectivas estructuradas facilitando la recopilación de comentarios sobre lo que se debe continuar haciendo, lo que

se debe detener y nuevas ideas que se pueden implementar para mejorar el proceso del marco Scrum.

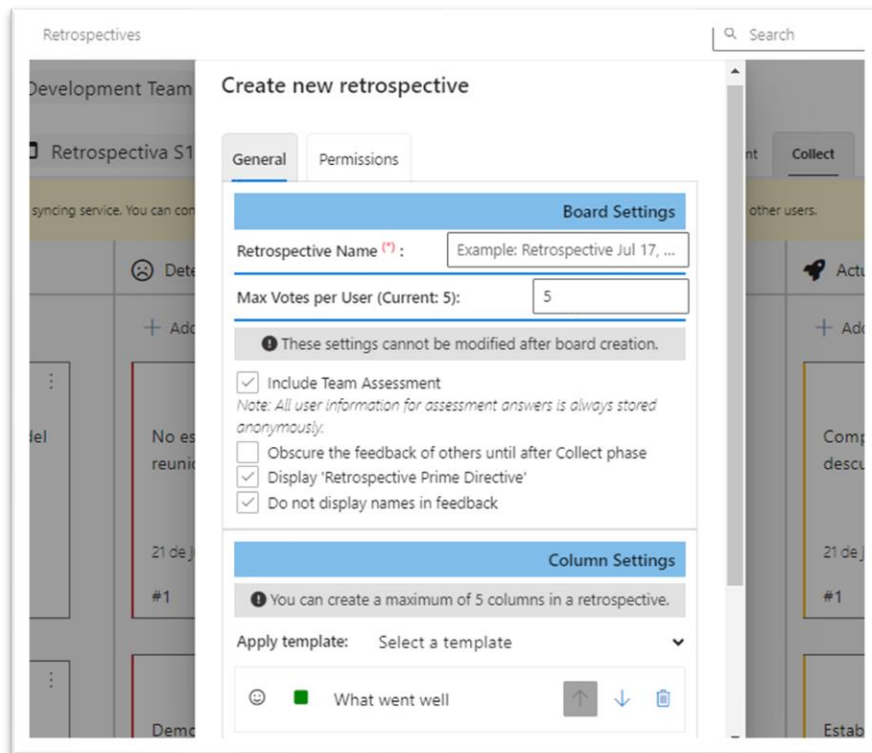


Figura 5. Creación de una nueva Retrospectiva con el plugin de Retrospective

Después de la entrega del primer release del producto, se configuró el plugin de “Estimate” en Azure Boards para mejorar el proceso de estimación durante las planificaciones de los siguientes sprints. Por ello fue necesario activar dicha extensión en el proyecto desde el Marketplace de Azure Boards y posterior habilitar los permisos necesarios para utilizarla.

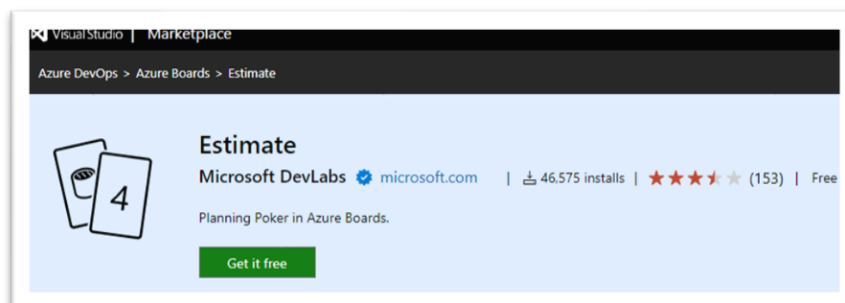


Figura 6. Extensión “Estimate” en el Marketplace de Azure DevOps

La extensión facilitó la creación de sesiones de estimación colaborativas en tiempo real que permitía en un inicio seleccionar las historias de usuario que se van a estimar a partir de los elementos de trabajo presentes en el backlog y posterior a ello cuando la sesión de trabajo iniciaba, el equipo de desarrollo podía determinar una estimación en puntos de

historia en base a la dificultad de la historia de usuario. Estas puntuaciones se discutían conjuntamente hasta llegar a un consenso, y la estimación final se asignaba directamente al elemento de trabajo.

2.3 Planificación del Producto

En la planificación del producto se tomaron en cuenta tres aspectos claves para definir una hoja de ruta clara: la visión del producto, la lista priorizada de alto nivel y el Product roadmap.

- Área de Valor de los stakeholders

Definir las áreas de valor de los stakeholders permite alinear sus intereses y necesidades con la visión del producto. En la Tabla 8 ilustra cómo el Gestor de Tareas Optiplan enfoca sus características hacia ciertas áreas de valor, afirmando que el producto no solo cumple con los requisitos básicos del mercado, sino que también se distingue y optimiza en términos de recursos.

Tabla 8. Áreas de valor de los Stakeholders para el Gestor de tareas Optiplan

Área	Característica	Descripción
Condiciones de Entrada	Paridad con la competencia	El sistema garantiza que se cumplen las características mínimas requeridas para competir en el mercado.
Facilitador	Facilita la entrada a un nuevo mercado	Permite a las organizaciones públicas optimizar la planificación operativa y mejorar la transparencia.
Diferenciador	Diferenciación de competidores	La gestión de tareas y flujo de trabajo basada en roles jerárquicos proporciona una estructura organizativa específica que se diferencia de herramientas genéricas como Excel o Trello.
Reducción de costos	Reducción de recursos	Facilita la supervisión y gestión integral de tareas, permitiendo una mejor distribución de los recursos humanos.

- Visión del Producto

Para definir la visión del producto Gestor Operativo Optiplan se utilizó la plantilla de Moore (Geoffrey A. Moore, 1999). A continuación, se presenta la visión del producto:

Para las entidades públicas **que** buscan gestionar su planificación operativa, reducir tiempos de entrega y aumentar la transparencia de sus proyectos, el **Gestor Operativo Optiplan** es una aplicación web para administrar las tareas operativas mediante tableros Kanban **que** facilita el flujo de las tareas mediante un enfoque visual y colaborativo **a diferencia** de las hojas de cálculo en Microsoft Excel y otros softwares genéricos de planificación como Jira, Trello o Asana.

Nuestro producto permite una gestión y supervisión integral de tareas desde la gerencia hasta el nivel operativo, con funcionalidades como el control del flujo de trabajo, notificaciones, carga de archivos a la tareas y seguimiento continuo del progreso de las tareas del proyecto.

- Product Backlog de Alto Nivel

Se presenta un panorama general de los requisitos y funcionalidades que se desarrollarán en el producto. En la Tabla 9, se detalla la descripción de las historias de usuario de alto nivel (épicas) del producto.

Tabla 9. Product backlog de Alto Nivel


#	Título	Descripción
1	Gestión Integral de Tableros Kanban	Como usuario quiero tener una gestión integral de mis tableros Kanban para organizarlos de manera lógica.
2	Gestión Integral de Tareas	Como usuario quiero gestionar tareas de los proyectos para controlar su flujo a través de las columnas del tablero Kanban.
3	Control y Priorización de Tareas	Como usuario quiero tener una vista completa y ordenada de las tareas pendientes en cada proyecto para priorizar y gestionar el trabajo.
4	Control y Seguimiento del Flujo de Trabajo	Como usuario quiero utilizar un gestor visual de tareas para evaluar visualmente el progreso de las tareas.
5	Sistema de Notificaciones	Como usuario quiero recibir notificaciones sobre la finalización de tareas con algún comentario adicional para controlar el progreso y los cambios en el proyecto.
6	Gestión de Documentos	Como usuario quiero poder cargar archivos a las tareas para tener una documentación de respaldo.
7	Acceso Basado en Roles	Como administrador quiero un acceso basado en roles jerárquicos para asegurar que cada usuario tenga permisos adecuados según su rol.

8	Generación de Reportes	Como usuario quiero generar reportes y visualizar gráficas sobre el progreso y desempeño del proyecto para tomar decisiones informadas.
---	------------------------	---

- Product Roadmap

Para definir una hoja de ruta inicial para la construcción del producto se utilizó el formato definido por Kenneth Rubin (Rubin, 2012). A continuación, se presenta el Product Roadmap, el cual detalla los objetivos, características mínimas liberables, la arquitectura y el calendario de las versiones planificadas para el Gestor de Tareas Optiplan.

Tabla 10. Product Roadmap del Gestor Operativo Optiplan

	2023-B	2024-A
		
Objetivo	Implementar las funcionalidades principales del gestor de tareas basado en Kanban y el acceso basado en roles jerárquicos.	Implementar la gestión de tareas y agregar funcionalidades avanzadas como reportes y notificaciones.
Características mínimas liberables	<ul style="list-style-type: none"> - Tablero Kanban - Flujos de tareas en Kanban - Acceso al sistema - Gestión de tareas de un proyecto. 	<ul style="list-style-type: none"> - Gestión de tareas por roles -Control y visualización del flujo de tareas por roles - Sistema de notificaciones. - Gestión de documentos.
Arquitectura	Aplicación Web	Aplicación Web
Calendario	1.0	2.0

2.4 Planificación del Release

Para determinar la planificación del Release, se utilizó el enfoque de fecha fija descrito en el texto Essential Scrum (Rubin, 2012). Se estableció que se realizarán dos (2) releases con una duración de siete (7) sprints cada uno, donde la fecha de entrega del producto está contemplada para el 4 de Julio del 2024. En la Tabla 11 y Tabla 12 se presentan las fechas de inicio y fin del Release 1 y 2 respectivamente.

Tabla 11. Fechas de inicio y fin de los sprints del Release 1

	Fecha inicio	Fecha fin
RELEASE 1	16-11-2023	29-02-2024
Sprint 1	16-11-2023	29-11-2023
Sprint 2	30-11-2023	14-12-2023
Sprint 3	15-12-2023	04-01-2024
Sprint 4	05-01-2024	18-01-2024
Sprint 5	19-01-2024	01-02-2024
Sprint 6	02-02-2024	15-02-2024
Sprint 7	16-02-2024	29-02-2024

Tabla 12. Fechas de inicio y fin de los sprints del Release 2

	Fecha inicio	Fecha fin
RELEASE 2	26-03-2024	04-07-2024
Sprint 8	26-03-2024	11-04-2024
Sprint 9	12-04-2024	25-04-2024
Sprint 10	26-04-2024	09-05-2024
Sprint 11	10-05-2024	23-05-2024
Sprint 12	24-05-2024	06-06-2024
Sprint 13	07-06-2024	20-06-2024
Sprint 14	21-06-2024	04-07-2024

Para calcular la capacidad del equipo durante estos sprints, se determinó la velocidad del equipo en un rango de **20 a 28** puntos de historia por sprint. Luego, se procedió a organizar el backlog del producto, donde se crearon, estimaron y priorizaron los elementos necesarios. Después, se tomó la velocidad más baja del equipo (18 puntos) y se multiplicó por el total de sprints (7), marcando así una línea en el backlog que señalaba los elementos que "deberán" completarse. De manera similar, se multiplicó la velocidad más alta (26 puntos) por el total de sprints, trazando una segunda línea en el backlog para indicar los elementos que "podrían tener" si el equipo trabaja al ritmo más rápido esperado. Esto implica que el equipo de desarrollo debería poder completar entre **126 y 182** puntos de historia durante cada lanzamiento.

A continuación, en la Tabla 13 se presenta el backlog del producto para el Release 1, con rangos de puntos de historia determinados por la planificación del release bajo el enfoque de fecha fija. Las historias de usuario están divididas en dos secciones: "WILL HAVE" (tendrán), que incluye las características esenciales, y "MIGHT HAVE" (podrían tener), que contiene características adicionales que se podrían completar si el equipo mantiene una velocidad más alta. Además, se ha señalado una línea que separa las características

mínimas liberables, etiquetada como “MUST HAVE”. Esta línea se encuentra sobre el primer rango de puntos de historia e indica un posible escenario positivo donde las características imprescindibles se completarán previo al fin del Release 1.

Tabla 13. Product Backlog bajo el enfoque fecha fija correspondiente a la planificación del Release 1

Numero Historia	Descripción	Puntos de Historia
HU01	Como usuario, quiero crear un tablero Kanban, proporcionando un nombre, descripción y estableciendo configuraciones básicas como columnas iniciales.	13
HU02	Como usuario, quiero poder editar la configuración de un tablero existente, incluyendo la adición, modificación o eliminación de columnas, la actualización de la descripción y otras preferencias.	8
HU03	Como administrador, quiero tener control sobre quién puede acceder y realizar cambios en los tablero Kanban para definir perfiles y permisos específicos.	21
HU05	Como usuario, quiero gestionar los proyectos de mi organización, para asegurarme de que las actividades y objetivos empresariales se cumplan.	13
HU06	Como usuario, quiero visualizar el listado de proyectos para conocer los proyectos de mi organización.	8
HU07	Como usuario, quiero buscar los proyectos por nombre; y ordenar ascendente y descendente sus características.	8
HU08	Como usuario, quiero gestionar las tareas de un proyecto para controlar el flujo de trabajo.	13
HU09	Como usuario, quiero buscar las tareas por nombre, estado y proyecto; y ordenar las características de las tareas de forma ascendente y descendente.	8
HU10	Como usuario, quiero definir las fechas de inicio y de fin de las tareas para optimizar la planificación y el seguimiento del cronograma de un proyecto.	5

HU11	Como usuario, quiero gestionar el flujo de las tareas dentro de un tablero Kanban para representar fácilmente su progreso y mantener el estado actualizado de cada tarea.	13
HU12	Como usuario, quiero priorizar las tareas dentro de un tablero Kanban, para visualizar las tareas más importantes y urgentes en primera posición.	13
WILL HAVE ^ 126 PUNTOS DE HISTORIA		
HU13	Como superior, quiero iniciar el flujo de trabajo de una tarea asignada a un subordinado para que este pueda visualizarla y comenzar a trabajar en ella.	13
HU14	Como usuario, quiero visualizar y subir archivos en las tareas de un proyecto para poder supervisar el progreso y tener acceso fácil a la información relevante.	21
HU15	Como usuario, quiero visualizar y enviar comentarios en las tareas de un proyecto para facilitar la comunicación entre los miembros de la organización.	8
HU16	Como superior, quiero controlar la carga de trabajo de mis subordinados para garantizar que no se generen cuellos de botella y se mantenga un flujo de trabajo eficiente.	13
MIGHT HAVE ^ 182 PUNTOS DE HISTORIA		
HU17	Como superior, quiero refinar mis tareas asignadas para dividir las en tareas más pequeñas y asignarlas a mis subordinados.	13
HU18	Como usuario, quiero recibir notificaciones cuando haya actualizaciones relevantes en el progreso de mis tareas para tomar acciones rápidas y mantenerme informado.	21
HU04	Como usuario, quiero poder organizar mis tableros Kanban en categorías o grupos para facilitar su acceso y gestión.	13
HU19	Como usuario, quiero tener acceso a un historial de cambios en la configuración del tablero Kanban para rastrear las modificaciones realizadas por mí o por otros miembros del equipo.	8



HU20	Como usuario, quiero tener la capacidad de definir estados personalizados para las tarjetas en cada columna, adaptando el flujo de trabajo a mis procesos específicos.	8
-------------	--	---

La Tabla 14 muestra el backlog del producto para el Release 2, bajo el mismo enfoque de fecha fija presentado anteriormente. De igual forma, la línea que separa las características mínimas liberables, etiquetada como “MUST HAVE” se encuentra sobre el primer rango de puntos de historia e indica un posible escenario positivo de cumplimiento del release 2.

Tabla 14. Product Backlog bajo el enfoque fecha fija correspondiente a la planificación del Release 2

Numero Historia	Descripción	Puntos de Historia
HU13	Como superior, quiero iniciar el flujo de trabajo de una tarea asignada a un subordinado para que este pueda visualizarla y comenzar a trabajar en ella.	13
HU14	Como usuario, quiero visualizar y subir archivos en las tareas de un proyecto para poder supervisar el progreso y tener acceso fácil a la información relevante.	21
HU15	Como usuario, quiero visualizar y enviar comentarios en las tareas de un proyecto para facilitar la comunicación entre los miembros de la organización.	8
HU16	Como superior, quiero controlar la carga de trabajo de mis subordinados para garantizar que no se generen cuellos de botella y se mantenga un flujo de trabajo eficiente.	13
HU17	Como superior, quiero refinar mis tareas asignadas para dividir las en tareas más pequeñas y asignarlas a mis subordinados.	13
HU18	Como usuario, quiero recibir notificaciones cuando haya actualizaciones relevantes en el progreso de mis tareas para tomar acciones rápidas y mantenerme informado.	21
HU21	Como usuario, quiero gestionar las tareas según mi rol, para asegurarme de que estoy cumpliendo con mis responsabilidades y contribuyendo al proyecto.	13

HU22	Como usuario, quiero visualizar el flujo de tareas según mi rol, para entender mejor mi carga de trabajo y priorizar mis actividades.	13
HU20	Como usuario, quiero definir estados personalizados para las tarjetas en cada columna para adaptar el flujo de trabajo a mis procesos específicos.	8
WILL HAVE ^ 126 PUNTOS DE HISTORIA		
HU03	Como usuario, quiero organizar mis tableros Kanban en categorías o grupos para facilitar su acceso y gestión.	13
HU19	Como usuario, quiero acceder a un historial de cambios en la configuración del tablero Kanban para rastrear las modificaciones realizadas por mí o por otros miembros del equipo.	8
HU23	Como usuario, quiero visualizar estadísticas clave sobre el desempeño general del tablero Kanban para seleccionar métricas específicas y periodos de tiempo.	21
MIGHT HAVE ^ 182 PUNTOS DE HISTORIA		
HU25	Como usuario, quiero que mi tablero Kanban se integre con herramientas de comunicación como Slack, Microsoft Teams o Discord para facilitar la notificación automática de cambios y actualizaciones.	21
HU26	Como usuario, quiero que las fechas límite y plazos en mi tablero Kanban se sincronicen automáticamente con calendarios externos como Google Calendar o Microsoft Outlook para mantener una visión integrada de las actividades.	21



2.5 Planificación del Sprint

La Tabla 15 muestra los objetivos definidos para cada sprint en la construcción del producto. Estos objetivos garantizan un progreso constante y enfocado hacia la entrega final, permitiendo al equipo trabajar en tareas prioritarias y acorde a sus capacidades.

Tabla 15. Tabla de objetivos por sprint

SPRINT	OBJETIVO
Sprint 1	Crear tableros Kanban
Sprint 2	Crear flujos de tareas en Kanban

Sprint 3	Controlar el acceso a los tableros Kanban
Sprint 4	Mejorar el flujo de trabajo de los tableros Kanban
Sprint 5	Crear proyectos para las tareas operativas
Sprint 6	Mejorar los flujos de trabajo de las tareas operativas
Sprint 7	Visualizar proyectos y tareas operativas
Sprint 8	Descubrir el flujo de tareas para todos los roles
Sprint 9	Gestionar las tareas del Gestor Empresarial
Sprint 10	Visualizar el flujo de tareas del Gestor Empresarial
Sprint 11	Gestionar las tareas del Gestor de Área
Sprint 12	Visualizar el flujo de tareas del Gestor de Área
Sprint 13	Gestionar las tareas del Líder de Área
Sprint 14	Gestionar la ejecución de tareas del Técnico

En el ANEXO I se presentan una captura de los objetivos por sprint redactados en la sección de la Wiki de Azure DevOps.

2.6 Desarrollo de Sprints

2.6.1 Sprint 1: Crear tableros Kanban

El **objetivo** del Sprint 1 estuvo relacionado a la creación de tableros Kanban en base a las configuraciones como el número de columnas y nombres de estas. La **velocidad** del equipo, al ser el primer sprint, se definió en 25 puntos de historia recomendado por el Scrum Master.

Las **historias de usuario** seleccionadas para el sprint fueron: la HU-01 relacionada con la creación de tableros Kanban y la HU-02 sobre la edición de la configuración del tablero Kanban, como se muestra en la Figura 7. Adicional, se agregó una historia de usuario técnica relacionada a la configuración del entorno y herramientas de desarrollo.

Order	Title	Assigned To	State	Priority	Story Points
1	HU01 Crear un Nuevo Tablero Kanban	CHRISTIAN PAT...	Closed	1	13
2	HU02 Editar Configuración del Tablero Kanban	CHRISTIAN PAT...	Closed	2	5
3	Configurar el Entorno Desarrollo	DAVID SEBASTI...	Closed	1	8

Figura 7. Sprint Backlog del Sprint 1

Para la estimación de las historias de usuario se utilizó la herramienta de Planning Poker Online, por su accesibilidad en línea, simplicidad y facilidad de uso, lo que facilita la colaboración y participación del equipo.

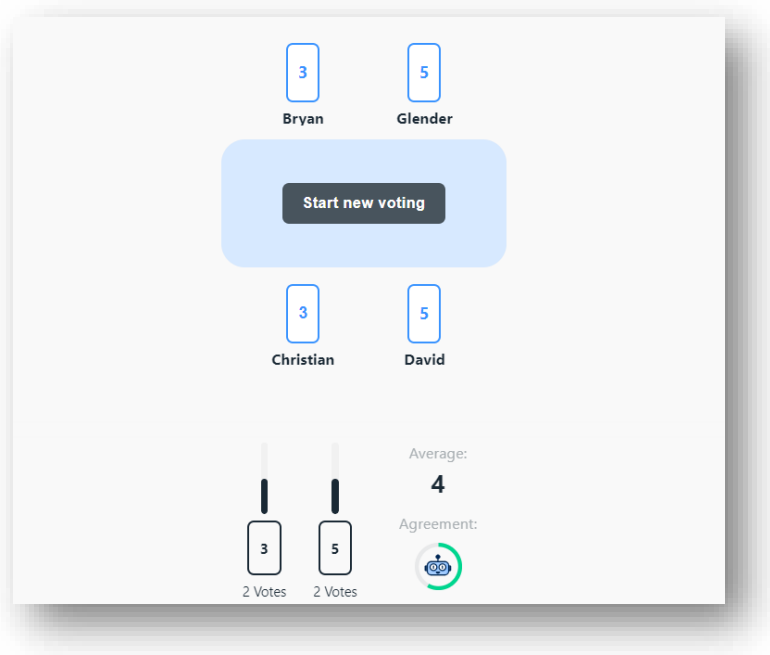


Figura 8. Estimación de las historias de usuario en el Planning Poker Online

La **capacidad** como equipo para la planificación de tareas durante este sprint de dos semanas con una duración 10 días laborales fue un total de 80 horas, donde cada miembro del equipo se asignó una capacidad de dos horas por día, tal como se observa en Figura 9.

User	Days off	Activity	Capacity per day
BRYAN MARCELO TAPIA VERGARA	0 days	Development	2
CHRISTIAN PATRICIO PAZMINO B.	0 days	Development	2
DAVID SEBASTIAN LEON OCHOA	0 days	Development	2
GLENDER ISRAEL MIRANDA CHAV	0 days	Development	2
Team days off	0 days	These days off apply to the whole team	

Figura 9. Capacidad del equipo para el sprint 1

Durante este Sprint 1, el enfoque del equipo fue en la historia de usuario HU-01, que consistía en la capacidad de crear un tablero Kanban. Esta historia de usuario fue priorizada como alta y se completó exitosamente durante el sprint. También se trabajó en la historia de usuario HU-02, que involucraba la funcionalidad de editar la configuración de un tablero existente, incluyendo la adición, modificación o eliminación de columnas, la actualización de la descripción y otras preferencias. Si bien el progreso de esta HU fue significativo, el equipo no fue capaz de completarla dentro del sprint asignado.

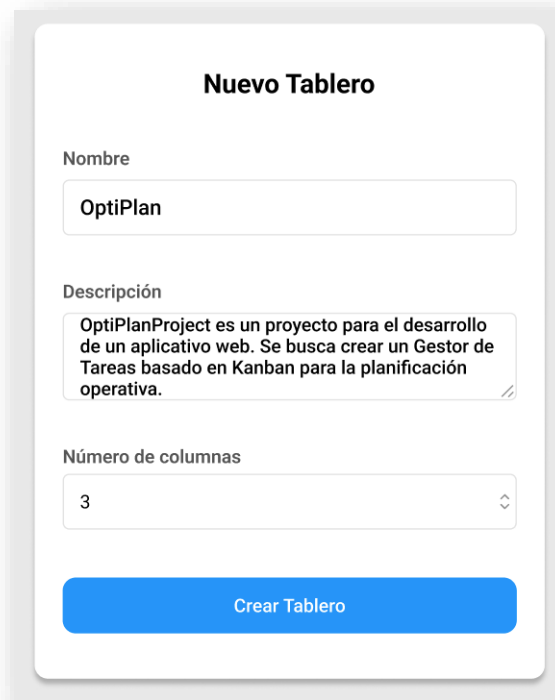
El horario para las sesiones diarias se eligió en consenso con el equipo, optando por la noche debido a los horarios de clases y trabajo. Durante las primeras **Dailys Scrum** se identificó que el tiempo de las reuniones se extendía mucho más allá de los 15 minutos recomendados. Esto se debía a que se abordaban temas que debían discutirse en otros momentos, lo que afectaba la eficiencia de las reuniones. Como parte de la adaptación, a mitad del sprint, se propuso el uso de la herramienta Miro para funcionar como una pizarra colaborativa (ver Figura 10). Esta herramienta facilitó la dinámica de las reuniones, permitiendo que fueran más ágiles y se plasmara la evidencia del trabajo realizado durante el día, mejorando la transparencia y participación durante las Daily Scrum.



Figura 10. Pizarra Colaborativa para la Daily Scrum

En la **Revisión** del Sprint, se realizó una demostración de la creación de un tablero Kanban. El Product Owner pudo inspeccionar el incremento de software (ver Figura 11). Así mismo, pudo evaluar el progreso de la HU-02 donde corroboró que no cumplía con todos los criterios de aceptación y por ende no se podía dar como finalizada dicha historia de usuario.

Durante la adaptación, movió esta historia de usuario para el próximo sprint con la indicación de que sea prioridad.



Nuevo Tablero

Nombre

OptiPlan

Descripción

OptiPlanProject es un proyecto para el desarrollo de un aplicativo web. Se busca crear un Gestor de Tareas basado en Kanban para la planificación operativa.

Número de columnas

3

Crear Tablero

Figura 11. Incremento del Sprint 1

Durante la **Retrospectiva** del Sprint, el equipo se enfocó en varios temas cruciales para mejorar el proceso de desarrollo del proyecto, en un ejercicio de reflexión y ajuste. Se debatió cómo optimizar la presentación del producto a los interesados, subrayando la importancia de una comunicación clara y efectiva. Se identificó la necesidad de crear una Definición de Terminado para establecer los parámetros que determinen cuándo un entregable está completo y listo para ser presentado. Además, se analizó la razón por la cual no se pudo completar la segunda historia de usuario, descubriendo obstáculos al

descomponer las historias de usuario en tareas técnicas. Como parte del ajuste, el Scrum Master ofreció una clase sobre cómo realizar este proceso de división de tareas.

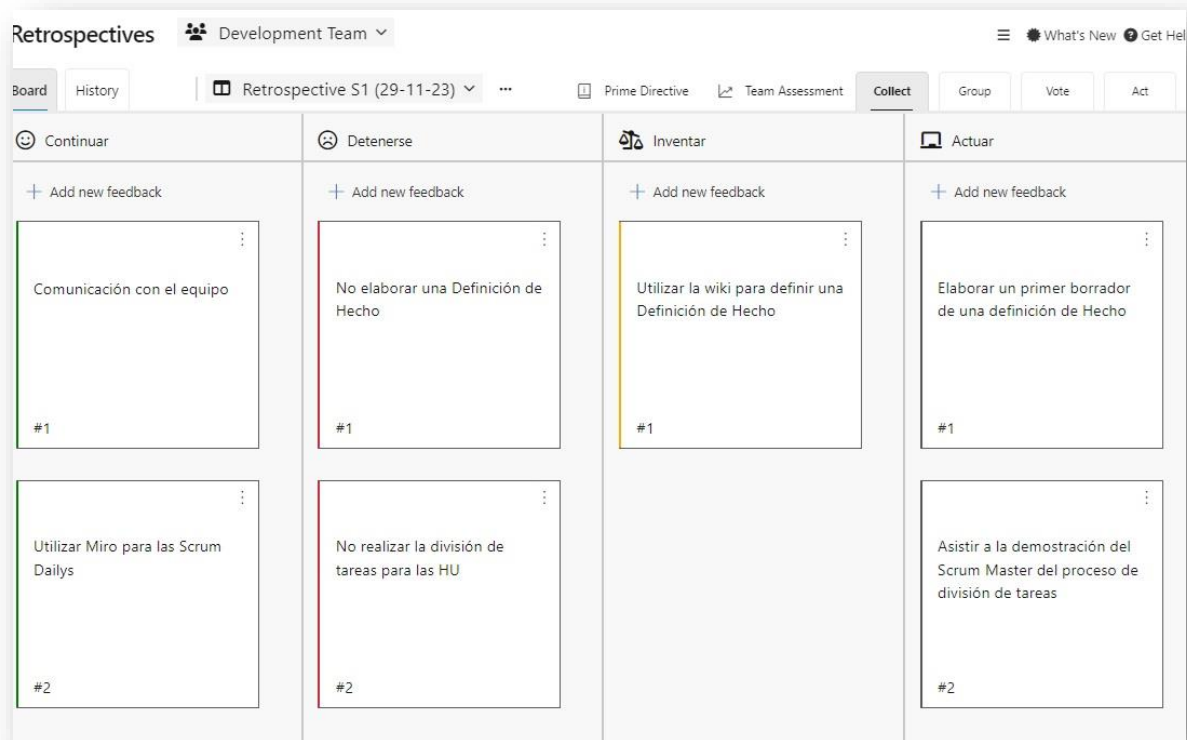


Figura 12. Retrospectiva del Sprint 1

2.6.2 Sprint 2: Crear flujos de tareas en Kanban

El **objetivo del Sprint 2** estuvo relacionado a replicar el flujo de tareas en los tableros Kanban. La **velocidad** del equipo se mantuvo en 25 puntos de historia.

Las historias de usuario seleccionadas para el Sprint 2 (ver Figura 13) fueron: nuevamente la HU-01 debido a que se agregaron nuevas tareas en base al feedback provisto por el Product Owner; la HU-02 porque no se completó en su totalidad en el anterior sprint. Adicionalmente, se agregó una historia de usuario técnica relacionada con la configuración de un entorno de producción, cuyo objetivo era utilizar dicho entorno como medio de presentación de los incrementos potencialmente entregables durante las revisiones del producto software. Asimismo, se creó la historia técnica relacionada con la creación del primer borrador para la Definición de Terminado.

+ -		Order	Title	Assigned To	State	Priority	Story Points
		1	> Creación de un DOD	CHRISTIAN PAT...	Closed	2	5
+		2	> HU01 Crear un Nuevo Tablero Kanban	CHRISTIAN PAT...	Closed	1	13
+		3	> HU02 Editar Configuración del Tablero Kanban	CHRISTIAN PAT...	Closed	2	5
		4	> Configurar el Entorno de Despliegue	GLENDER ISRA...	Closed	1	8

Figura 13. Sprint Backlog del Sprint 2

Durante las **Dailys Scrum** de este sprint, se observó una notable reducción en el tiempo de las reuniones, logrando adherirse a los 15 minutos recomendados. La reducción del tiempo de las reuniones representó una carga menor para el equipo, permitiendo ocupar dicha brecha de tiempo en la adaptación de obstáculos que surgían durante la inspección.

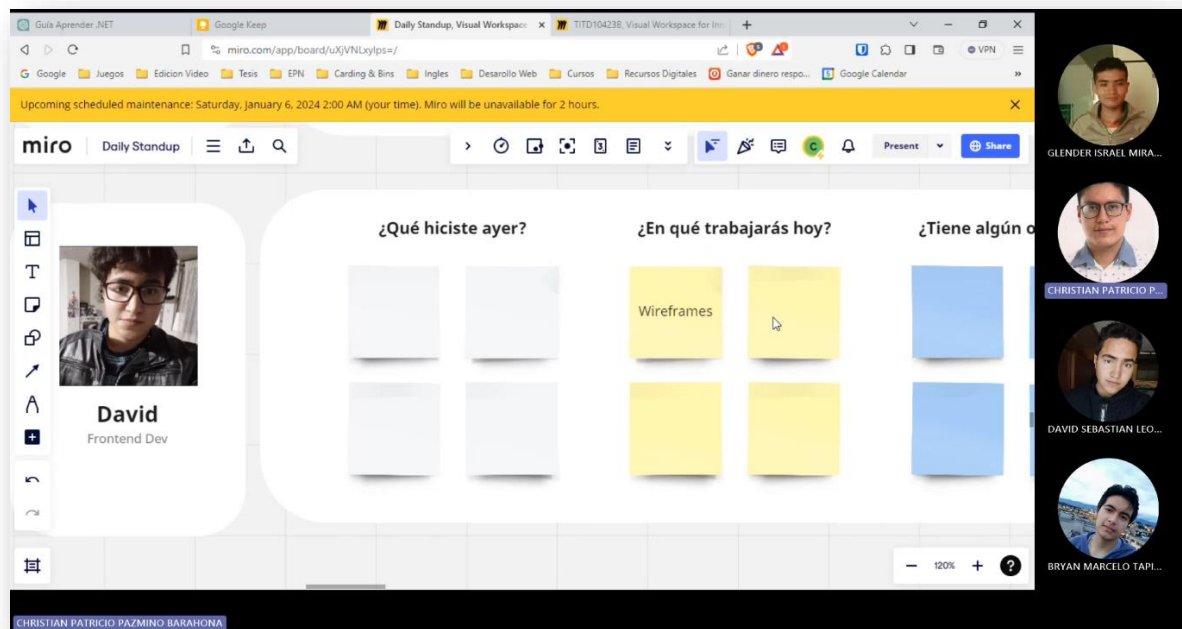


Figura 14. Scrum Daily en Microsoft Teams

El incremento de software presentado al equipo durante la **Revisión** (Ver Figura 15) sorprendió de forma positiva al Product Owner y Scrum Master, debido a que se habían cumplido con todas las historias de usuario planificadas y verificaba que el entregable generaba valor al usuario final.

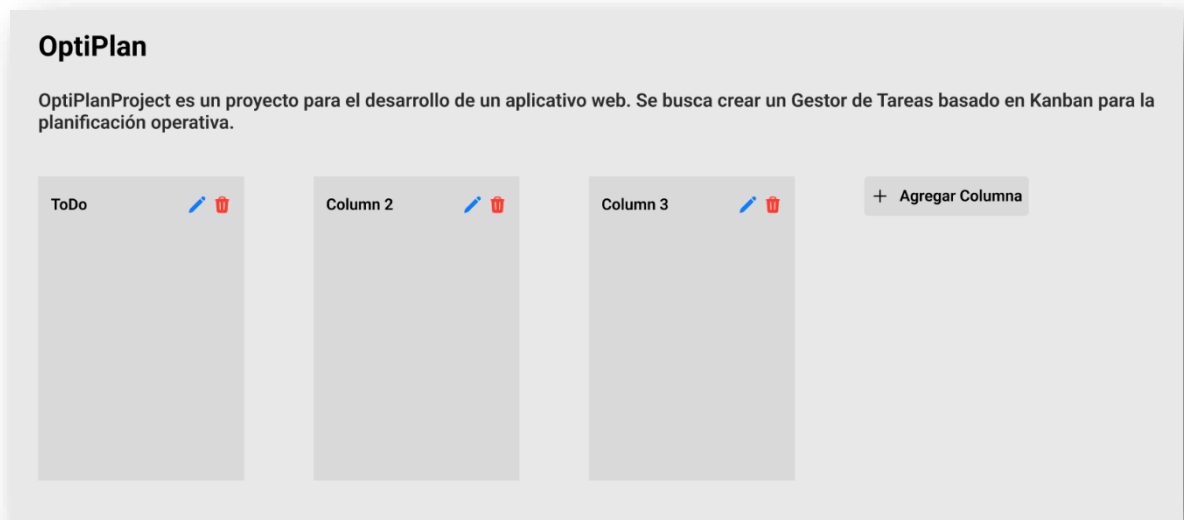


Figura 15. Incremento del Sprint 2

En la fase de inspección de la **Retrospectiva**, se identificaron aspectos positivos como la asistencia del Scrum Master a las Daily Scrum y la necesidad de completar completamente las historias de usuario. Se detectaron problemas importantes, como la falta de una Definición de Terminado que sea transparente y verificable, y la ausencia de un periodo de tiempo específico para dividir las historias de usuario en tareas técnicas. Además, se señaló la carencia de una presentación preparada para la demostración durante la revisión del producto.

En respuesta a estos hallazgos, en la fase de adaptación, se propusieron varias acciones. Se decidió redactar la Definición de Terminado en las wikis del proyecto para garantizar su claridad y transparencia. Asimismo, se sugirió realizar una presentación en la herramienta de Canva previo a las sesiones de revisión del producto para asegurar que toda la información sobre el contexto del sprint sea presentada al equipo Scrum.

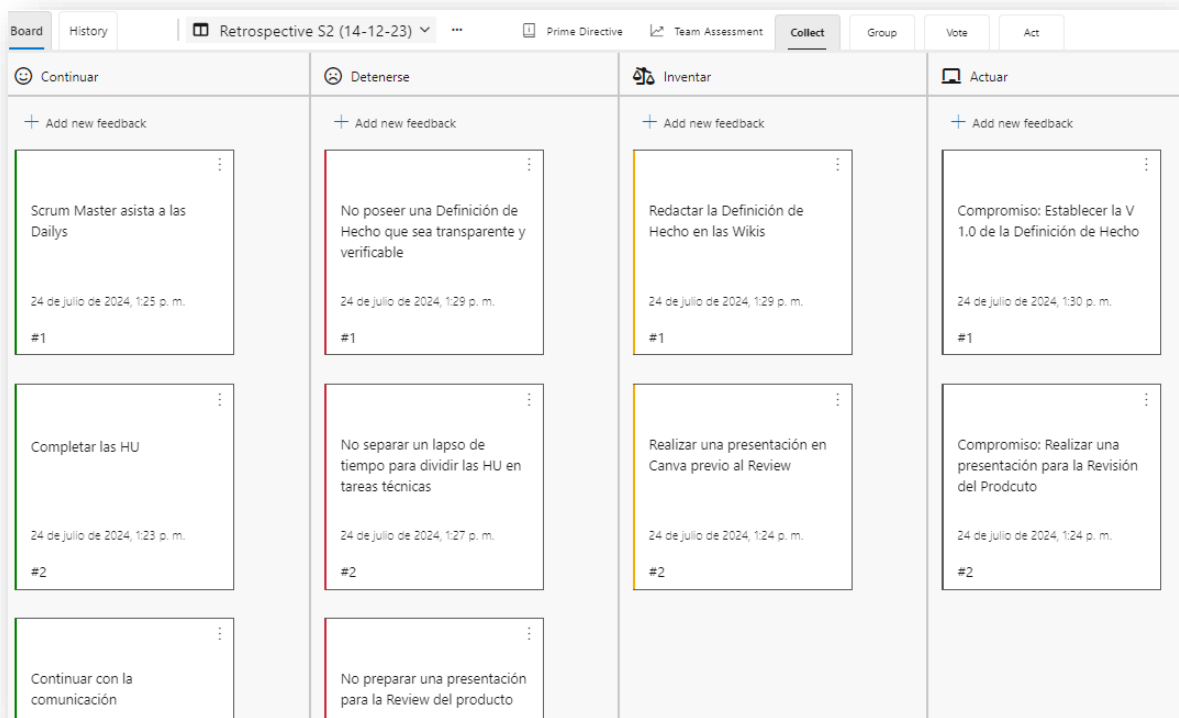


Figura 16. Retrospectiva del Sprint 2

2.6.3 Sprint 3: Controlar el acceso a los tableros Kanban

El **objetivo del Sprint 3** abarcaba el método de acceso al sistema y los privilegios que debía poseer el usuario para visualizar sus tableros Kanban. La **velocidad** del equipo se mantuvo en 25 puntos de historia.

La única **historia de usuario** seleccionada para el sprint 3 fue: la HU-03 relacionada con el método de acceso y autenticación del sistema. Esta HU, que implícitamente incluye el requisito no funcional del sistema de acceso para el usuario, fue priorizada debido a la complejidad que conlleva construir dicha funcionalidad.

Order	Title	Assigned To	State	Priority	Story Points
1	H03 Establecer Permisos de Acceso al Tablero	CHRISTIAN PAT...	Closed	1	21
	Actualización del contrato de la API	BRYAN MARCE...	Closed	2	
	Creación y Actualización Mock API	BRYAN MARCE...	Closed	1	
	Crear Modelo Usuario	BRYAN MARCE...	Closed	2	
	Crear el formulario Login	DAVID SEBASTI...	Closed	1	
	Controlador con los métodos (API)	DAVID SEBASTI...	Closed	2	
	Crear Formulario Register	CHRISTIAN PAT...	Closed	1	
	Establecer los Guards y Decorators para la aut...	BRYAN MARCE...	Closed	2	
	Definir casos de prueba	BRYAN MARCE...	Closed	2	
	Proteger las rutas de la API	BRYAN MARCE...	Closed	2	
	Definir los esquemas para JWT	BRYAN MARCE...	Closed	2	
	Servicio para el registro y login de usuarios	BRYAN MARCE...	Closed	2	

Sprint Backlog del Sprint 3

Algo a destacar durante algunas **Dailys Scrum** fue que el Scrum Master estuvo presente en la inspección de dichas reuniones, lo que permitió a posterior durante las adaptaciones se puedan resolver dudas técnicas u obstáculos de forma inmediata.

En el sprint anterior, se asumió el compromiso de establecer una Definición de Terminado en las Wikis de Azure DevOps. Para mantener un control adecuado de las versiones, se decidió emplear una nomenclatura numérica. En la Figura 17, se presentan los ítems que conforman la versión 1.0 de la Definición de Terminado.

DEFINICIÓN DE TERMINADO	
Versión: 0.0.1	
Autores: Equipo de Desarrollo	
Objetivo: Garantizar la calidad, transparencia y entrega consistente del software desarrollado al establecer criterios claros y compartidos que definen cuándo un producto de Software se encuentra terminado, promoviendo así la eficiencia del equipo y la satisfacción del cliente en el proyecto web.	
Categoría	Item
Código	<p>El código cumple con las reglas de sintaxis establecidas en el documento 'Estándar de Codificación para JavaScript' versión 0.0.1 elaborado por el equipo de desarrollo.</p> <p>El código ha superado todas las verificaciones de ESLint sin errores ni advertencias, alcanzando un 0% de problemas identificados.</p>
Control de Versiones	Los commits han sido escritos siguiendo el estándar definido en el documento 'Buenas Prácticas para escribir commits' versión 0.0.1 elaborado por el equipo de desarrollo.
Historias Usuario	Se ha evaluado el entregable para confirmar su cumplimiento con los criterios de aceptación establecidos.
Pruebas	El código ha pasado por pruebas unitarias con la meta de alcanzar un 100% de éxito en su ejecución.
Despliegue	<p>Todos los pipelines automatizados en Azure han sido ejecutados con éxito, asegurando así un 100% de cumplimiento en la integración continua y en las pruebas automatizadas.</p> <p>El sistema se encuentra correctamente desplegado.</p>
Documentación	Se han actualizado los manuales de usuario para reflejar los nuevos cambios.

Figura 17. Definición de terminado - Versión 1.0

Durante la ejecución del Sprint, las discusiones entre los Developers y el Product Owner permitieron una comprensión más profunda de las necesidades y expectativas de los usuarios, proporcionando una visión renovada de las funcionalidades de la aplicación. Un hallazgo clave fue la identificación de cinco perfiles de usuario, cada uno con responsabilidades específicas que afectan su interacción con el sistema: Gestor Empresarial (GE), Gestor de Área (GA), Líder de Área (LA), Técnico y Administrador del sistema.

En la inspección de la **Revisión** del Producto se presentó el incremento de software (Ver Figura 18), el cual tuvo fallos de calidad al inicio de la exposición, esto debido a que previamente no se había corroborado de que el posible entregable cumpliera con todos los criterios de la Definición de Terminado. Esto origino que se perdiera calidad en el incremento y por ende disminuyera el valor del producto de software hacia el usuario.

Nuevo Usuario

Nombre
Roberto García

Correo
roberto.garcia@epn.edu.ec

Contraseña

Perfil
Gestor Empresarial

Departamento
TI

Crear Usuario

Figura 18. Incremento del Sprint 3

Durante la **Retrospectiva** del Sprint, se recalcó la importancia de la comunicación por WhatsApp y la asistencia a las Daily Scrum como elementos positivos a mantener. Sin embargo, se detectaron problemas, como realizar despliegues a producción a última hora y esperar hasta el final del sprint para resolver problemas técnicos. En la fase de adaptación, se propusieron algunas mejoras como: validar con el Product Owner los criterios de la Definición de Hecho previo a la revisión; refinar en conjunto las historias de usuario comprometidas para el sprint durante los primeros días, esto incluía definir criterios de aceptación durante las conversaciones. Otra medida a adoptar fue publicar hilos en Teams cuando ocurría un problema técnico que el Developer no podía solucionar fácilmente, el objetivo era asegurar un mejor seguimiento y soporte efectivo.

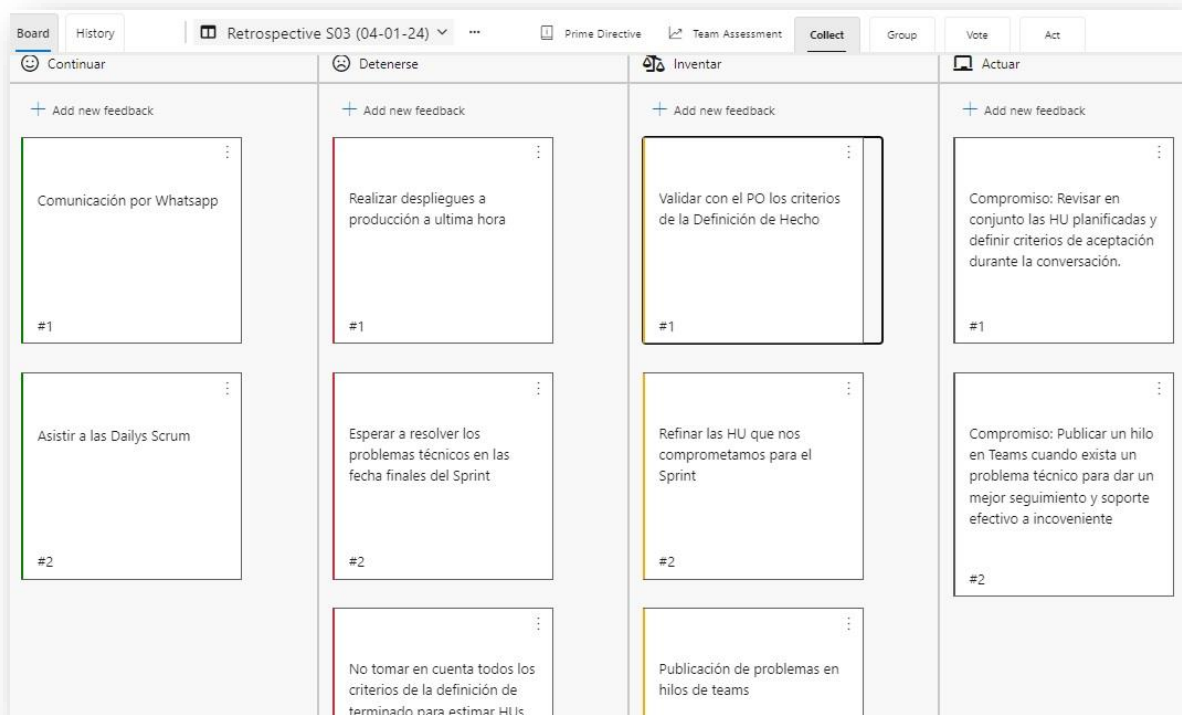


Figura 19. Retrospectiva del Sprint 3

2.6.4 Sprint 4: Mejorar el flujo de trabajo de los tableros Kanban

El **objetivo** del Sprint 4 se centró en mejorar el flujo de trabajo de los tableros Kanban. Se buscó optimizar la gestión de tareas, implementar mejoras en la visualización y facilitar la colaboración entre los miembros del equipo. La **velocidad** del equipo, basada en la experiencia acumulada de sprints anteriores, se estableció en 21 puntos de historia. Durante este sprint, se priorizaron las historias de usuario que permitieran una mayor eficiencia en el uso de los tableros Kanban.

Las **historias de usuario** seleccionadas para el sprint 4 fueron: la HU-05 relacionada con optimizar el flujo de las tareas dentro un tablero Kanban de forma general; la HU-06 relacionada con la gestión de tareas en un backlog para el perfil de mayor jerarquía, el Gestor Empresarial.

Order	Title	Assigned To	State	Priority	Story Points
1	HU06 Gestión de tareas GE	DAVID SEBASTI...	Closed	1	13
	Crear Menu Desplegable lateral	DAVID SEBASTI...	Closed	2	
	Actualizar el API Contract	BRYAN MARCE...	Closed	2	
	Modificar la arquitectura de las tareas	BRYAN MARCE...	Closed	1	
	Consumir API para crear tarea en Backlog	GLENDER ISRA...	Closed	2	
	Consumir API para obtener los nombres de us...	DAVID SEBASTI...	Closed	3	
	Manejar el Token en LocalStorage	GLENDER ISRA...	Closed	2	
	Agregar el servicio para gestionar las tareas d...	BRYAN MARCE...	Closed	2	
	Generar datos "seed" para usuarios, tableros y...	BRYAN MARCE...	Closed	1	
2	HU05 Mejorar el Flujo de Tareas	DAVID SEBASTI...	Closed	1	8

Figura 20. Sprint Backlog del Sprint 4

El desarrollo de las **Dailys Scrum** no presentaron ningún inconveniente relevante durante la ejecución del Sprint. La única sugerencia que surgió fue la posibilidad de que el resto de Developers accediera también a la pizarra colaborativa de Miro y pudieran llenar por sí mismos las tarjetas, promoviendo así una mayor colaboración entre los miembros del equipo.

En la **Revisión** del Sprint se inspeccionó el incremento de software sin un contexto previo sobre el desarrollo del Sprint, lo que ocasionaría a posterior una posibilidad de mejora. La satisfacción de los interesados con el entregable de este sprint fue baja. Aunque hubo avances, no representaron un valor considerable para el usuario final. Esto llevó a una revisión crítica del enfoque del equipo para asegurar que los futuros desarrollos se alineen mejor con las necesidades y expectativas del usuario. Durante la adaptación, el Scrum Master hizo un comentario acerca del diseño de las interfaces de la aplicación web, proponiendo la necesidad de crear un sistema de diseño para mejorar la coherencia visual y estandarización de componentes y estilos.

The screenshot shows the OptiPlan Backlog interface. On the left, there are navigation options for 'Backlog' and 'Tablero'. The main area is titled 'Backlog' and contains a table of tasks. A 'Crear tarea' button is located in the top right corner of the table area.

Nombre	Responsable	Estado	Fecha de inicio	Fecha de fin	Acciones
Establecer un plan de trabajo	Juan Guerrero	New	2024-07-14	2024-07-19	✎ 🗑
Identificar y contratar proveedores	María García	New	2024-07-20	2024-08-10	✎ 🗑
Realizar un inventario	María García	New	2024-07-12	2024-08-03	✎ 🗑
Clasificar las reparaciones	Juan Guerrero	New	2024-07-19	2024-07-25	✎ 🗑
Realizar una evaluación	Glender Miranda	New	2024-07-21	2024-07-28	✎ 🗑
Establecer un presupuesto	David Pazmiño	New	2024-07-27	2024-08-04	✎ 🗑

Figura 21. Incremento del Sprint 4

En la **Retrospectiva** del Sprint se identificaron prácticas que deberían continuar, detenerse y mejorarse. Entre las prácticas a continuar destacaron la publicación de problemas en hilos de Teams y el apoyo mutuo entre los miembros del equipo. Sin embargo, se detectaron varias prácticas que debían detenerse, se señaló la presentación de la Review de manera no estructurada y sin contexto, y la falta de preparación adecuada para la presentación. También se mencionó la falta de compromiso con la planificación, estimación y priorización, así como no verificar los criterios de aceptación y dejar las pruebas de despliegue para el final del sprint. Otra mala práctica fue la acumulación de tareas para el final lo que impidió la de toma de decisiones oportunas en áreas que requerían ajustes. En la fase de adaptación, se propusieron varias acciones para abordar estos problemas: se sugirió implementar datos "semilla" para las pruebas y priorizar la calidad del aplicativo. Además, se destacó la importancia de cumplir con la Definición de Hecho y estimar las horas de las tareas en la planificación del Sprint. Se propuso definir un sistema de diseño para la aplicación, aplicando estrategias de otras empresas para lograr consistencia visual y reducir problemas de interfaz de usuario (UI por sus siglas en inglés) al final del proceso de desarrollo. Finalmente, después de la votación por las mejores soluciones, se materializaron en elementos de trabajo la posibilidad de utilizar un nuevo framework para el desarrollo del frontend y la creación de branches o pipelines de desarrollo para mejorar el proceso de construcción de software.

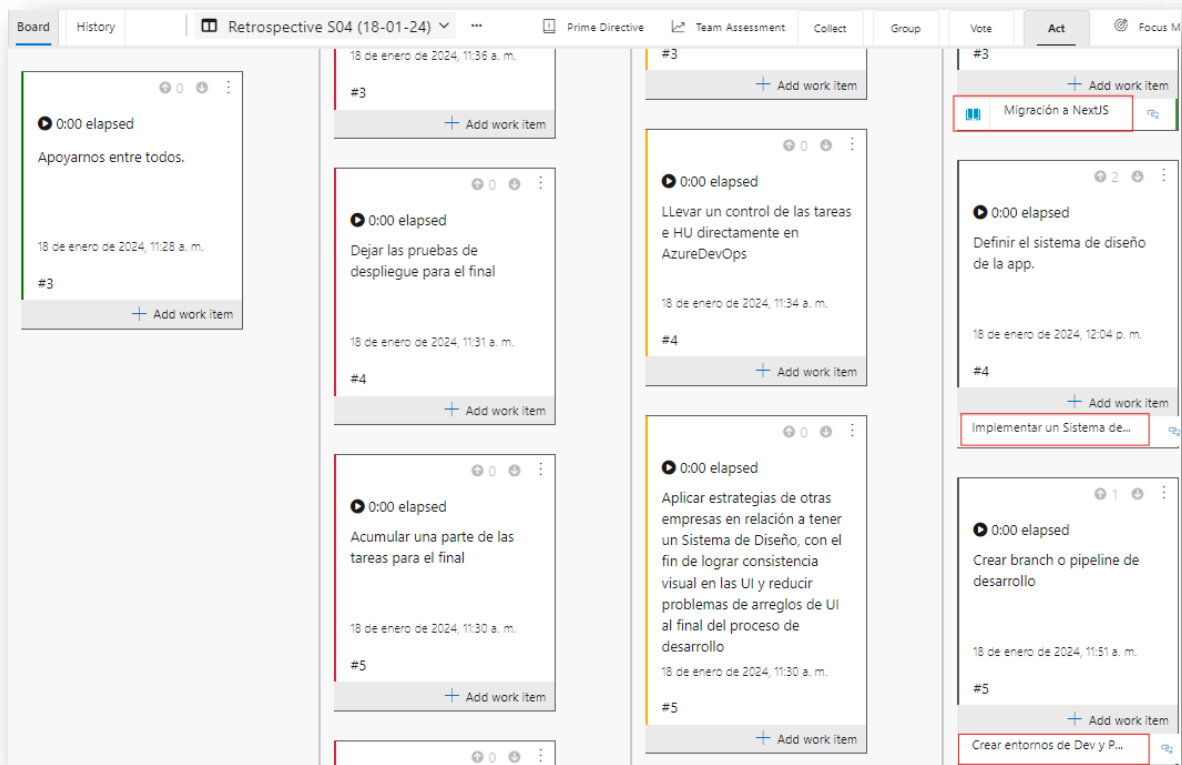


Figura 22. Retrospectiva del Sprint 4

2.6.5 Sprint 5: Crear proyectos para las tareas operativas

El **objetivo** del Sprint 5 fue desarrollar la funcionalidad que permite la creación de proyectos destinados a gestionar las tareas operativas. Este sprint se centró en la implementación de características que permiten a los usuarios crear, editar y organizar proyectos, así como asignar tareas específicas a estos proyectos. La **velocidad** del equipo, basada en la experiencia de sprints anteriores, se estableció en 25 puntos de historia.

Las **historias de usuario** seleccionadas para el Sprint 5 fueron: la HU-07, centrada en permitir a los usuarios crear y visualizar nuevos proyectos; la HU-08, que permite al perfil de gestor empresarial visualizar los proyectos de su organización; y la HU-03, reintroducida para actualizar los métodos de acceso y permisos debido a la historia técnica relacionada con la migración del framework del frontend. En cuanto a las historias técnicas, se implementó la migración a NextJs para mejorar la administración de tokens, y la creación de entornos de desarrollo y producción por separado, para automatizar cada entorno y mejorar el proceso de integración continua, facilitando así la implementación de prácticas DevOps que aseguren la entrega rápida y segura de los incrementos del producto.

Sprint 5						
Order	Title	Assigned To	State	Priority	Story	
1	> HU-07 Crear proyecto	BRYAN MARCE...	● Closed	2	8	
2	> Implementar un Sistema de Diseño	DAVID SEBASTI...	● Closed	2	8	
3	▼ H03 Establecer Permisos de Acceso al Tablero	CHRISTIAN PAT...	● Closed	1	21	
	📌 Consumir o diseñar un metodo de autentificaci...	CHRISTIAN PAT...	● Closed	1		
	📌 Agregar la entidad que relacione al usuario co...	BRYAN MARCE...	● Closed	2		
	📌 Modificar el módulo de autenticación de acue...	BRYAN MARCE...	● Closed	2		
4	> HU-08 Visualizar los proyectos	DAVID SEBASTI...	● Closed	1	8	
5	> Migración a NextJS - Investigación	DAVID SEBASTI...	● Closed	1	5	
6	> Crear entornos de Dev y Prod	GLENDER ISRA...	● Closed	2	8	

Figura 23. Sprint Backlog del Sprint 5

En las **Daily Scrum** se enfatizó la importancia de ser transparentes y honestos al inspeccionar los avances de cada developer. Esta práctica permitió identificar y resolver obstáculos de manera oportuna, garantizando que todos los miembros estén alineados con los objetivos del sprint.

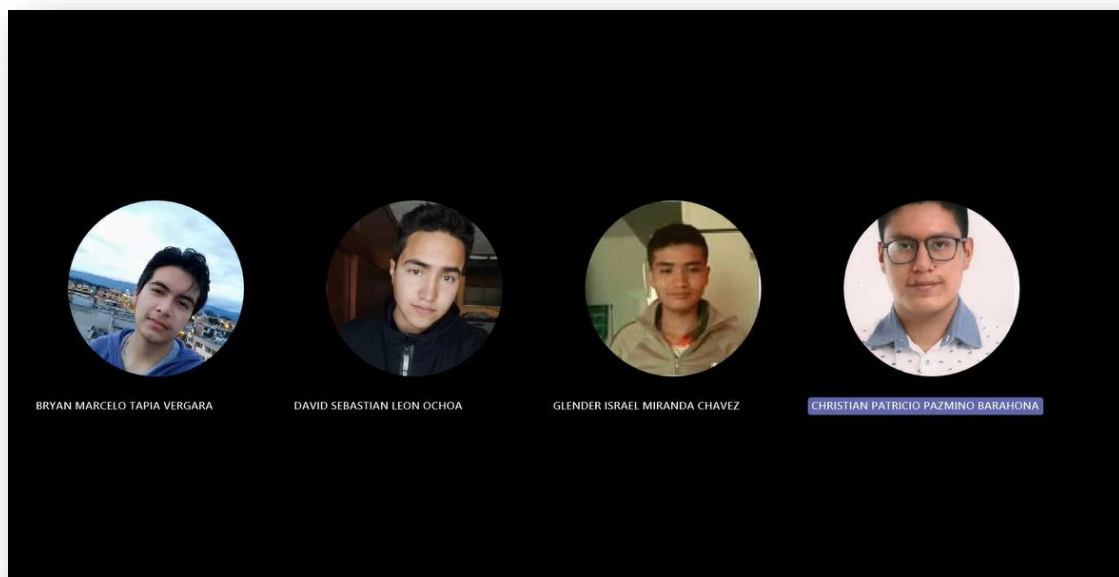
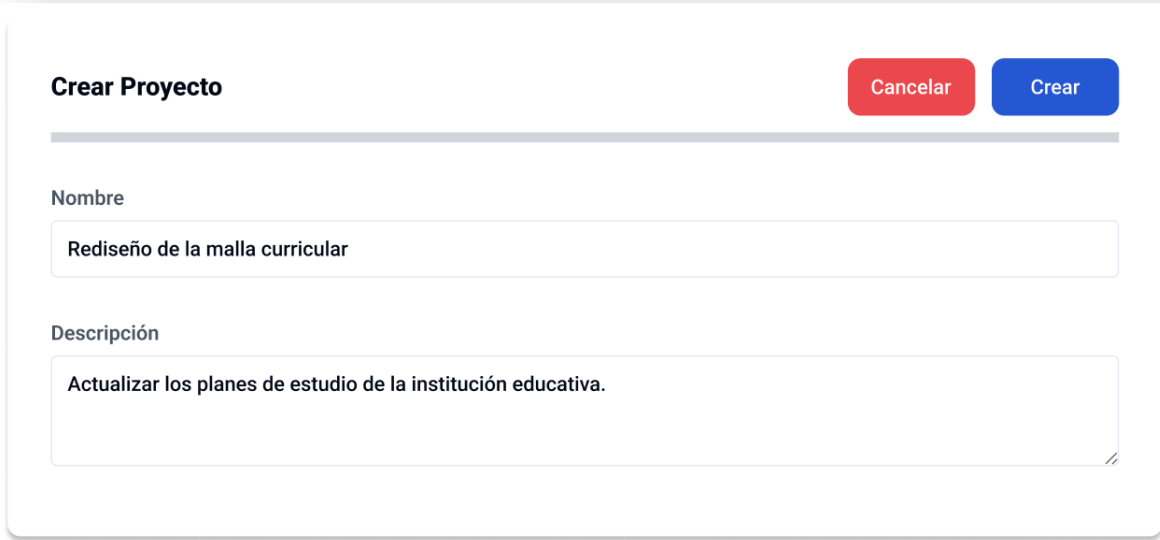


Figura 24. Daily Scrum en Microsoft Teams

Durante la **Revisión** del producto surgió un inconveniente con el entorno de producción: el tiempo de carga era excesivo y no reflejaba los cambios más recientes que se habían implementado unas horas antes. Debido a este problema, se decidió presentar el incremento de software utilizando el entorno de desarrollo. En respuesta a esta situación, el Product Owner señaló que el incremento de software no se podía considerar un entregable completo, ya que no cumplía con todos los criterios de la Definición de Hecho. En la adaptación, al revisar que algunas historias de usuario no se encontraban totalmente finalizadas, se tomó la decisión de trasladarlas al próximo sprint.



The image shows a web form titled "Crear Proyecto". At the top right, there are two buttons: "Cancelar" (red) and "Crear" (blue). Below the title, there is a horizontal separator line. The form contains two main sections: "Nombre" with a text input field containing "Rediseño de la malla curricular", and "Descripción" with a larger text area containing "Actualizar los planes de estudio de la institución educativa." The form has a clean, modern design with rounded corners and a white background.

Figura 25. Incremento del Sprint 5

Durante la **Retrospectiva** del Sprint 5, el equipo consideró importante continuar utilizando Azure Boards para gestionar historias de usuario y tareas técnicas, mantener una comunicación efectiva, fomentar el trabajo en equipo y registrar en Boards las horas esfuerzo que se hayan empleado para alcanzar el objetivo del sprint. Sin embargo, se detectaron varios problemas que debían abordarse. Entre ellos, iniciar trabajos que no se podían completar dentro del sprint, no priorizar la finalización de historias de usuario anteriores y dedicar mucho tiempo a implementaciones que generaban poco valor para el usuario final. También se identificó la acumulación de trabajo sin un seguimiento adecuado, realizar despliegues a producción a última hora y las estimaciones sin datos históricos como áreas que requerían mejoras inmediatas. En la fase de adaptación, se propusieron varias acciones para mitigar estos problemas. Se sugirió que los desarrolladores participen en reuniones semanales enfocadas en problemas técnicos del código y obstáculos en el desarrollo que no se han podido solucionar individualmente, con el objetivo de encontrar

soluciones rápidas y efectivas en equipo. Dichas reuniones también incluían una presentación inicial de los avances, similar a una revisión del producto, para fomentar la transparencia y la confianza dentro del equipo. Otra acción recomendada fue trabajar en una tarea a la vez, alineándose con el principio de WIP (Work In Progress) para mejorar la eficiencia y evitar sobrecargas de trabajo. Además, se recomendó realizar builds locales antes de subir los cambios al entorno de desarrollo, con el objetivo de identificar y corregir errores potenciales, asegurando así un despliegue más fluido y confiable.

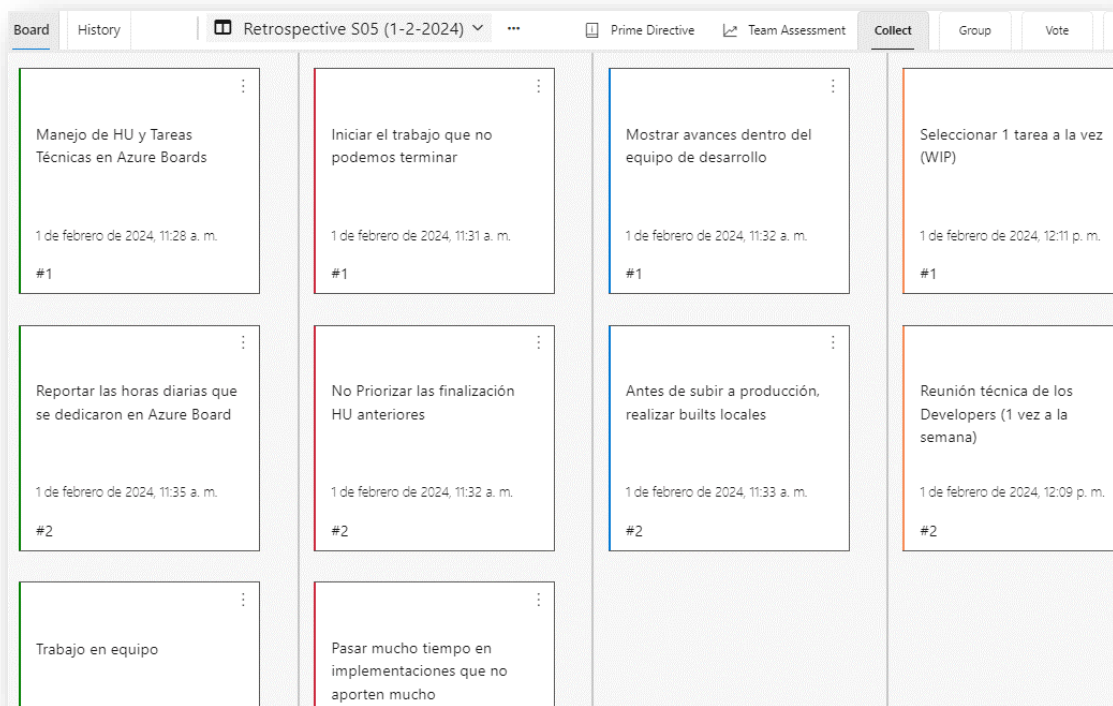


Figura 26. Retrospectiva del Sprint 5

2.6.6 Sprint 6: Mejorar los flujos de trabajo de las tareas operativas

El **objetivo** del Sprint 6 se centró en recrear el inicio del flujo de trabajo de una tarea que ha sido asignada a un miembro de la organización para que este pueda visualizarla y comenzar a trabajar en ella. La **velocidad** del equipo se redujo debido a la no culminación de historias de usuario en el sprint anterior, ajustándose en 18 puntos de historia.

Las **historias de usuario** seleccionadas para el Sprint 6 incluyeron varias tareas del sprint anterior que no se completaron y una nueva historia de usuario. Las HU-07, HU-06 y la tarea técnica de "Migración a NextJS - Implementación" se trasladaron al Sprint 6 debido a su incompletitud previa, lo que también resultó en una disminución en la estimación de

puntos de historia. La historia técnica "Migración a NextJS - Implementación" busca mejorar el rendimiento y la reutilización de componentes del frontend. La nueva historia de usuario, HU-09, se agregó para permitir a los gestores empresariales iniciar el flujo de trabajo de una tarea asignada a un subordinado, crucial para cumplir con el objetivo del sprint. Adicionalmente, se introdujo la historia técnica "Descubrimiento del producto sobre la gestión de tareas del sistema", que implica la creación de prototipos para comprender el flujo de trabajo de los perfiles de usuario: Gestor Empresarial, Gestor de Área, Líder de Área y Técnico.

Order	Title	Assigned To	State	Priority	Story Points
1	> [] HU-07 Crear proyecto	BRYAN MARCE...	Closed	2	3
2	> [] Descubrimiento del producto sobre la gestión de... DAVID SEBASTI...	DAVID SEBASTI...	Closed	1	5
3	> [] HU06 Gestión de tareas GE	DAVID SEBASTI...	Closed	1	3
4	> [] HU-09 Inicio del Flujo de una tarea	DAVID SEBASTI...	Closed	2	8
5	> [] Migración a NextJS - Implementacion	DAVID SEBASTI...	Closed	1	5

Figura 27. Sprint Backlog del Sprint 6

Un aspecto destacable durante las **Daily Scrum** fue la implementación de prácticas ágiles de desarrollo, especialmente el Pair Programming en la fase de adaptación. Esta técnica permitió al equipo resolver conflictos y errores de código de manera más eficiente, acelerando significativamente la finalización de tareas. Gracias a esta estrategia colaborativa, el equipo mejoró su eficiencia y productividad.

Para el incremento de software de este Sprint 6 (Ver Figura 28) se agregó valor a través de las confirmaciones a ciertas acciones, durante la inspección de la **Revisión** del producto, el Product Owner aprobó dichas funcionalidades y las calificó como un acierto.



Figura 28. Incremento del Sprint 6

Durante la **Retrospectiva** del Sprint 6 se consideró importante continuar con la práctica de realizar reuniones de presentación de avances los viernes o sábados, ya que estas ayudan a mantener una evaluación constante del progreso. También se destacó el uso de prácticas ágiles de desarrollo como el Pair Programming y si bien, la asistencia técnica no forma parte de las responsabilidades del Scrum Master, su presencia especialmente durante las dailys y reuniones técnicas fue de gran ayuda para el equipo de developers. Por otro lado, se identificaron varios problemas que debían abordarse. Entre ellos: la falta de comunicación hacia el equipo sobre los problemas en la Integración y entrega continua (CI/CD por sus siglas en inglés); el desarrollo del frontend sin un enfoque estructurado de programación. Además, se observó la necesidad de mejorar la comunicación entre los equipos de backend y frontend para evitar malentendidos y mejorar la coordinación.

En la fase de adaptación, se propusieron varias acciones para mitigar estos problemas. Se recomendó mejorar las prácticas DevOps, específicamente en CI/CD, y asegurarse de que cada tarea asignada a un miembro del equipo sea realizada en una rama separada que a posterior se unifique a la rama de desarrollo mediante pull requests. También se sugirió reestructurar los pipelines y releases para agilizar el despliegue y la entrega continua. Finalmente, se decidió adoptar un enfoque de programación basado en componentes para el desarrollo del frontend, con el objetivo de mejorar la modularidad y reutilización del código.

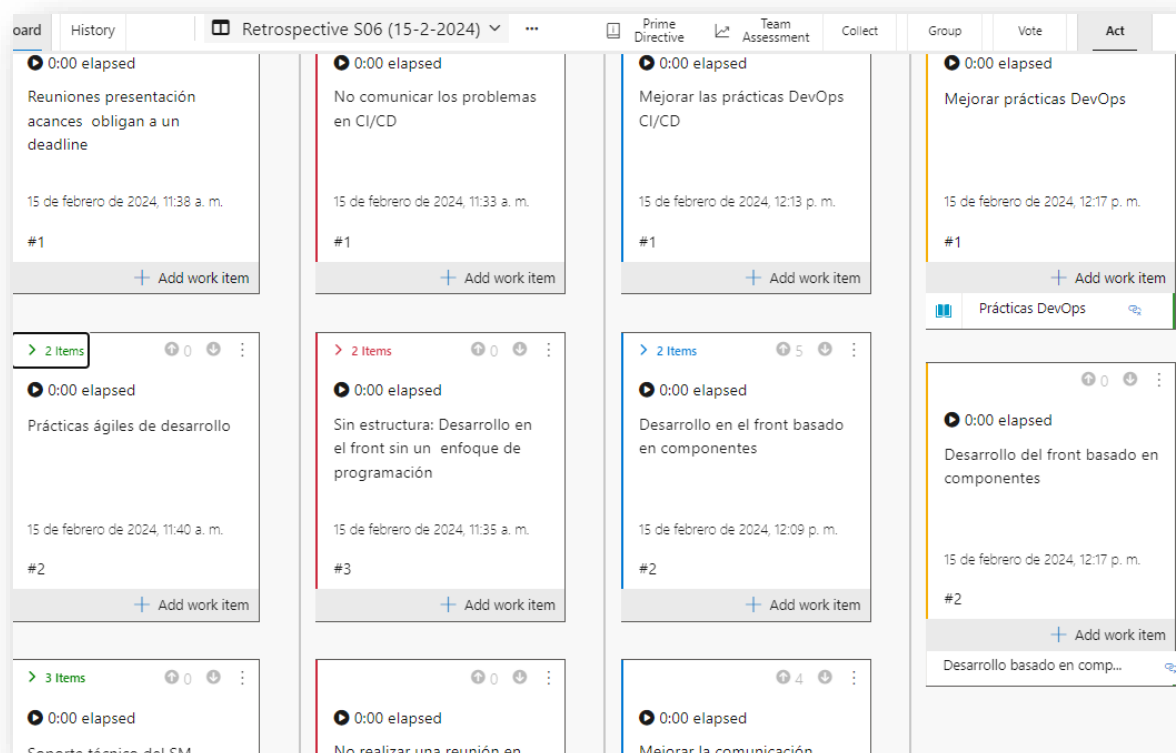


Figura 29. Retrospectiva del Sprint 6

2.6.7 Sprint 7: Visualizar proyectos y tareas operativas

El **objetivo** del Sprint 7 se enfocó en finiquitar las historias de usuario relacionadas con la visualización de proyectos y gestión de tareas operativas en el tablero Kanban para el perfil del gestor empresarial. Este sprint represento un momento importante en la construcción del producto software, ya que simbolizó la culminación y entrega de la Release 1. Por esta razón, se puso un énfasis especial en la calidad del software, dado que las características mínimas liberables ya habían sido implementadas. La **velocidad** del equipo, en base a la experiencia de anteriores sprint, se estableció en 24 puntos de historia.

Las **historias de usuario** seleccionadas para el Sprint 7 se centraron en finiquitar y completar el trabajo pendiente de sprints anteriores, con un enfoque principal en requisitos no funcionales como la documentación y pruebas unitarias. A excepción de la HU-10, todas las historias de usuario fueron continuaciones de esfuerzos previos. La HU-10, permite al perfil del gestor empresarial asignar personal responsable a sus tareas, facilitando el seguimiento del progreso.

Order	Title	Assigned To	State	Priority	Story
1	<ul style="list-style-type: none"> HU-10 Asignación de un responsable a una tarea Formulario de Actualizacion Consumir API 	CHRISTIAN PAT...	Closed	1	8
2	<ul style="list-style-type: none"> HU-07 Crear proyecto Pruebas Unitarias Documentar código 	BRYAN MARCE...	Closed	2	3
3	HU06 Gestión de tareas GE	DAVID SEBASTI...	Closed	1	3
4	HU-08 Visualizar los proyectos	DAVID SEBASTI...	Closed	1	3
5	HU-09 Inicio del Flujo de una tarea	DAVID SEBASTI...	Closed	2	1
6	<ul style="list-style-type: none"> Migración a NextJS - Investigación Migrar el modulo del Backlog 	DAVID SEBASTI...	Closed	1	5

Figura 30. Sprint Backlog del Sprint 7

Durante las **Daily Scrum** no se presentaron inconvenientes y su desarrollo fue fluido y satisfactorio para los developers.

La **Revisión** del sprint 7 se consideró un hito importante al ser la entrega del primer incremento de software, por ello los Scrum Developers expusieron detalladamente las características fundamentales del producto software desarrolladas hasta ese momento: el tablero Kanban, el flujo de tareas operativas en Kanban, el sistema de acceso al sistema a través de roles y la gestión de tareas al interior de un proyecto. Estas funcionalidades englobaron exitosamente las características mínimas liberables que se establecieron en la planificación del release, lo que se consideró un punto positivo para el progreso del proyecto. En Figura 31 se observa la pantalla de gestión de proyectos para el perfil del Gestor Empresarial.

Nombre	Porcentaje de avance	Fecha de inicio	Fecha de fin	Acciones
Software para gestión de TI	100%	2024-02-02	2024-03-03	[editar] [eliminar]
Proyecto de innovación	100%	2023-12-01	2024-12-11	[editar] [eliminar]
Preplanificación FIS EPN	100%	2024-05-09	2024-05-21	[editar] [eliminar]
Proyecto de mejoramiento de infraestructura	100%	2024-07-19	2024-08-25	[editar] [eliminar]
Establecimiento de un programa de energías renovables en las instalaciones.	100%	2024-07-31	2024-08-10	[editar] [eliminar]
Proyecto de inclusión digital	100%	2024-05-02	2024-05-25	[editar] [eliminar]
Reparación de la infraestructura	100%	2024-07-10	2024-08-25	[editar] [eliminar]
Renovación de laboratorios	100%	2024-07-25	2024-07-31	[editar] [eliminar]
Contratación de personal	100%	2024-07-04	2024-09-07	[editar] [eliminar]
Diseño e implementación de un plan de movilidad sostenible.	100%	2024-05-20	2024-05-24	[editar] [eliminar]

Figura 31. Incremento del Sprint 7 - Lista de proyectos

En la Figura 32 se observa la funcionalidad del tablero Kanban y el flujo de tareas operativas en el mismo.

Panel del proyecto

- Información
- Backlog
- Tablero

← Proyectos / Tablero / Proyecto de innovación

Por Hacer (5)	En Progreso (1)	Terminado (1)
<p>0%</p> <p>Identificación de Problemas y Oportunidades</p> <p>Etiqueta</p> <p>2024-05-30 - 2024-05-30</p> <p>3 2</p>	<p>50%</p> <p>Coordinar reuniones internas.</p> <p>Etiqueta</p> <p>2024-01-01 - 2024-02-02</p> <p>3 2</p>	<p>100%</p> <p>Revisión de prototipos</p> <p>Etiqueta</p> <p>2024-01-01 - 2024-02-02</p> <p>3 2</p>
<p>0%</p> <p>Desarrollar una investigación de mercado</p> <p>Etiqueta</p> <p>-</p> <p>3 2</p>		

Figura 32. Incremento del Sprint 7 – Tablero Kanban

Para la **Retrospectiva** del Sprint 7 se consideró importante continuar con la comunicación efectiva por varios canales, así como con el compromiso con lo justo y necesario, ya que estas prácticas han demostrado ser beneficiosas para el equipo. Sin embargo, se identificaron varios problemas que debían ser abordados. Entre ellos, la falta de asignación de recursos para el descubrimiento del producto con el Product Owner y la priorización inadecuada de historias que otorgan valor al usuario. Se reconoció la necesidad de

fomentar la conversación y entendimiento de los deseos del cliente y de crear un elemento de trabajo específico relacionado con el descubrimiento del producto.

En la fase de adaptación, se propusieron varias acciones para mitigar estos problemas. Se decidió diseñar prototipos de baja fidelidad para recoger retroalimentación temprana de los usuarios y dedicar el siguiente sprint a comprender de forma detallada el flujo de tareas para todos los roles del sistema.

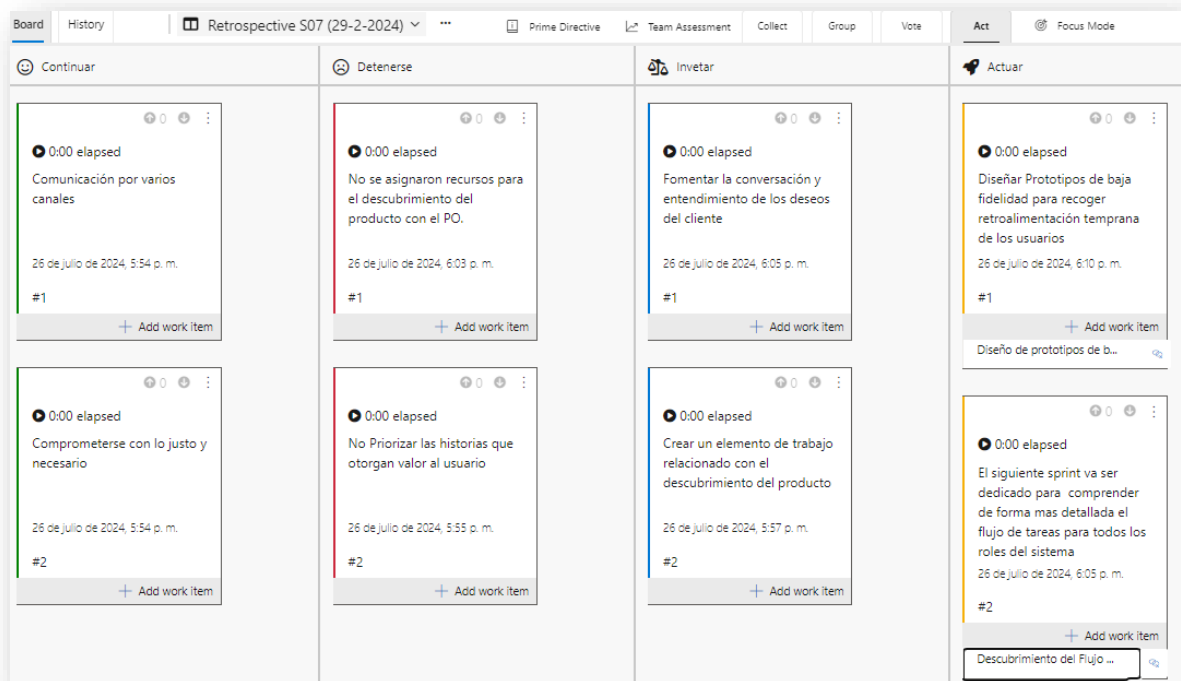


Figura 33. Retrospectiva del Sprint 7

2.6.8 Sprint 8: Descubrir el flujo de tareas para todos los roles

El **objetivo** del Sprint 8 se centró en descubrir y documentar el flujo de tareas para todos los roles del sistema. La **velocidad** del equipo se estableció en 28 puntos de historia posterior a una discusión y consenso entre el equipo.

Las **historias de usuario** seleccionadas para el Sprint 8 incluyeron varias historias técnicas como: el refinamiento del contrato de la API para garantizar una integración clara y eficiente entre los servicios; la creación de prototipos de baja fidelidad de todos los roles del sistema para recoger retroalimentación temprana de los clientes; la implementación de prácticas DevOps para mejorar el tiempo de entrega del incremento del producto software; y el descubrimiento y documentación del flujo de tareas de todos los roles del sistema para asegurar que cada rol tenga claras sus responsabilidades y actividades dentro del proceso.

Sprint 8						
Order	Title	Assigned To	State	Priority	Story P	
1	Refinar el Contrato de la API	BRYAN MARCE...	Closed	1	5	
	Refinamiento del Contrato en base a las neces...	BRYAN MARCE...	Closed	1		
	Documentar la API con Swagger	BRYAN MARCE...	Closed	2		
2	Diseño de prototipos de baja fidelidad	DAVID SEBASTI...	Closed	1	5	
	Diseñar los prototipos del sistema	DAVID SEBASTI...	Closed	2		
	Primer re diseño de los prototipos	DAVID SEBASTI...	Closed	2		
	Segundo re diseño de los prototipos	DAVID SEBASTI...	Closed	2		
3	Prácticas DevOps	GLENDER ISRA...	Closed	1	5	
	Pull Request	GLENDER ISRA...	Closed	2		
	Branches	GLENDER ISRA...	Closed	2		
	Releases	GLENDER ISRA...	Closed	2		
	Pipelines	GLENDER ISRA...	Closed	2		
4	Descubrimiento del Flujo de Tareas	CHRISTIAN PAT...	New	1	13	

Figura 34. Sprint Backlog del Sprint 8

Durante las **Daily Scrum**, no se presentaron inconvenientes y su desarrollo fue fluido y satisfactorio para los desarrolladores. El equipo se mantuvo enfocado en sus tareas, y la colaboración fue efectiva. La implementación de prácticas ágiles, especialmente el pair programming, ayudó a resolver conflictos y errores de código de manera eficiente, mejorando la productividad del equipo.

Aunque durante es Sprint no se generó un incremento de software debido a que su objetivo estaba relacionado exclusivamente con el descubrimiento del producto, durante la reunión de **Revisión** se presentaron a todo el equipo Scrum las interfaces y la explicación del flujo de tareas por cada uno de los roles.

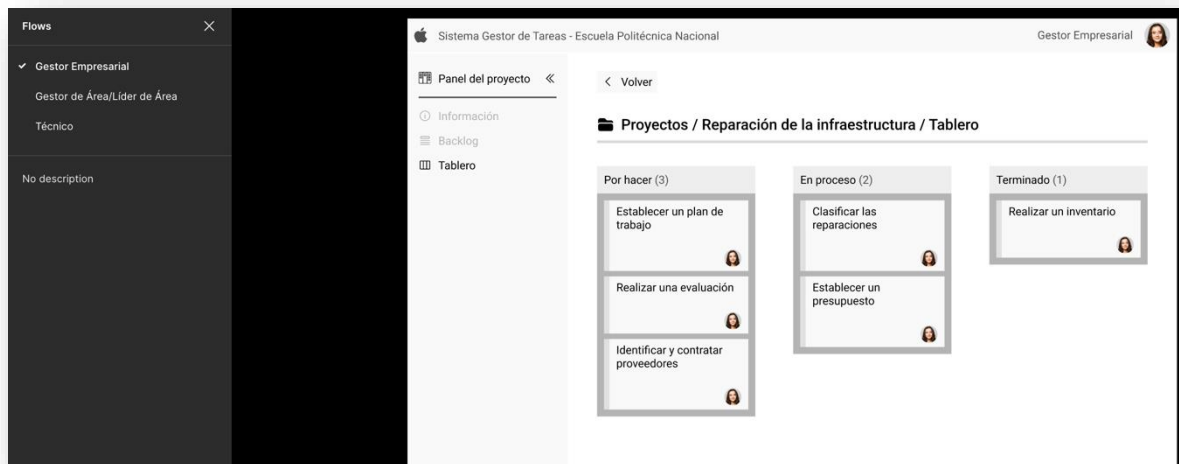


Figura 35. Prototipos para del Sprint 8

Durante la **Retrospectiva** del Sprint 8, se consideró importante continuar con el apoyo y la comunicación constante dentro del equipo y mantener una buena organización de las tareas en Azure. No obstante, se identificaron varios problemas que debían ser abordados. Entre ellos, la postergación de una tarea técnica relacionada con la organización de componentes en el frontend. También se propuso fomentar el uso de herramientas como Husky para aplicar reglas de formato y calidad de código en los commits.

En la fase de adaptación, se establecieron varias acciones para mitigar los problemas identificados. Se decidió definir un proceso de entrega continua a producción, refinar la definición de terminado, la creación de la definición de listo y establecer un horario fijo para documentar las horas de esfuerzo y avances en las tareas del Azure Board.

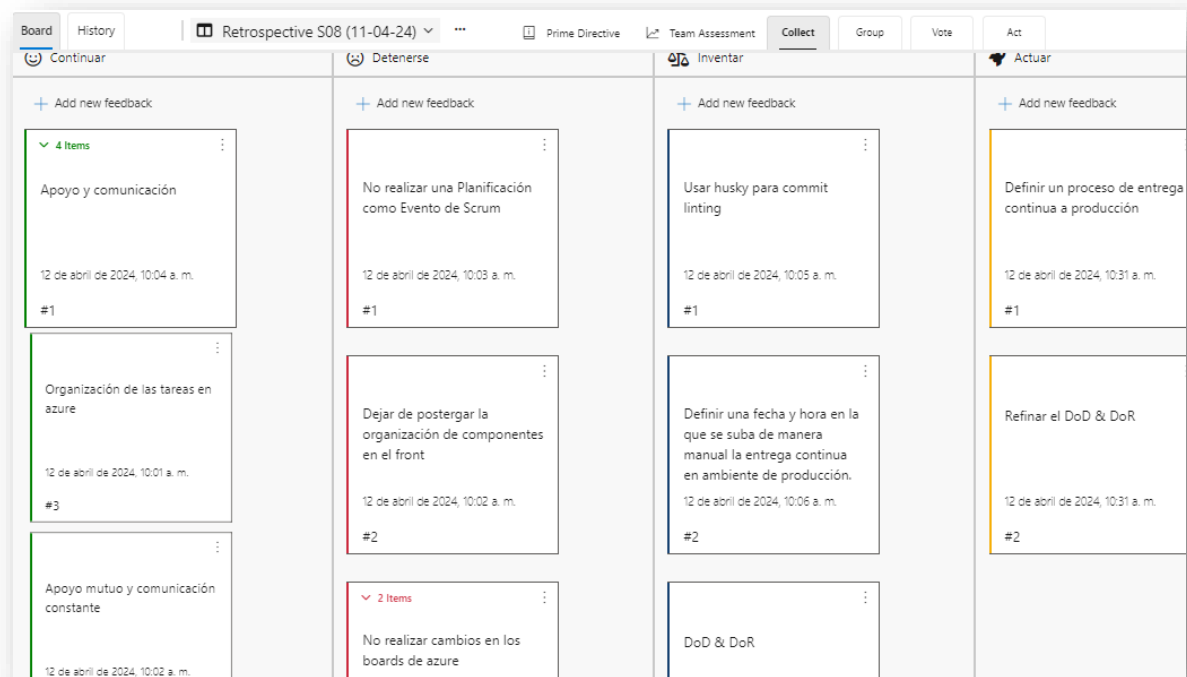


Figura 36. Retrospectiva del Sprint 8

2.6.9 Sprint 9: Gestionar las tareas del Gestor Empresarial

El **objetivo** del Sprint 9 se centró en gestionar las tareas del Gestor Empresarial. La **velocidad** del equipo disminuyó a 21 puntos de historia, basándose en la cantidad de puntos completados satisfactoriamente en los sprint anteriores.

Las **historias de usuario** seleccionadas para el Sprint 9 incluyeron la asignación de fechas de inicio y fin de una tarea, y de un proyecto en el caso del gestor empresarial, con el objetivo de optimizar la planificación y el seguimiento del cronograma del proyecto. También se incluyó el requisito no funcional relacionado con la gestión de sesión del usuario, en donde se debía desarrollar una interfaz nueva de inicio y cierre de sesión para que el gestor empresarial pueda acceder y salir de su cuenta personal. Además, se seleccionó la historia sobre la creación de un manual de usuario, solicitado por los stakeholders del proyecto, cuyo propósito es servir como guía de todas las funcionalidades del sistema. Otra historia de usuario fue la asignación de personal responsable a las tareas, lo cual permite al gestor empresarial asegurarse de que cada tarea tenga un encargado claro, facilitando el seguimiento del progreso de las tareas asignadas a sus subordinados. Adicionalmente, se propuso trabajar durante el sprint en un proceso de entrega continua a producción para que los stakeholders puedan verificar las funcionalidades del producto. Finalmente, se tomó en cuenta la historia técnica que proponía la refinación de la definición

de terminado (DoD por sus siglas en ingles) y elaboración de una definición de listo (DoR por sus siglas en inglés) con el propósito de asegurar que las funcionalidades cumplan con los estándares de calidad.



Order	Title	Assigned To	State	Priority	Story
1	> HU-11 Asignación de fechas	GLENDER ISRA...	● Closed	2	3
2	> Gestión de sesión del usuario	DAVID SEBASTI...	● Closed	2	5
3	> Manual de Usuario	CHRISTIAN PAT...	● Closed	2	3
4	> HU-10 Asignación de un responsable a una tarea	CHRISTIAN PAT...	● Closed	1	1
5	> Entrega Continua a Producción	GLENDER ISRA...	● Closed	1	3
6	> DoD & DoR	CHRISTIAN PAT...	● Closed	1	5

Figura 37. Sprint Backlog del Sprint 9

Durante las **Daily Scrum** del Sprint 9, hubo interrupciones significativas debido a problemas energéticos en el país que provocaron cortes de luz a nivel nacional. Esta situación imprevista impidió que el equipo se pudiera reunir de manera constante, lo que resultó en la cancelación de algunas sesiones y la realización de otras con solo la mitad de los integrantes. Estas inconsistencias afectaron gravemente la transparencia en el reporte de avances y limitaron la capacidad de adaptación, ya que no se pudieron discutir los ajustes necesarios con todos los miembros.

En el sprint anterior, se asumió el compromiso de establecer una Definición de Terminado en las Wikis de Azure DevOps. Para mantener un control adecuado de las versiones, se decidió emplear una nomenclatura numérica. En la Figura 38, se presentan los ítems que conforman la versión 2.0 de la Definición de Terminado. A diferencia de la primera versión, se incluyen prácticas detalladas para NextJS y NestJS, una mayor cobertura en pruebas unitarias, y una estandarización más estricta en los commits. Además, se refuerzan los criterios de aceptación de las historias de usuario y se establece la necesidad de actualizar

los manuales de usuario, lo que se traduce en la obligación de entregar un software de calidad.

Versión: 0.0.2
Autores: Equipo de Desarrollo
Objetivo: Garantizar la calidad, transparencia y entrega consistente del software desarrollado al establecer criterios claros y compartidos que definen cuándo un producto de Software se encuentra terminado, promoviendo así la eficiencia del equipo y la satisfacción del cliente en el proyecto web.

CÓDIGO

- El código sigue las reglas de sintaxis y mejores prácticas para NextJS y NestJS, además del estándar de codificación para JavaScript versión 0.0.1 del equipo de desarrollo.
- El código en el frontend y backend está debidamente comentado y documentado, incluyendo la documentación de componentes y servicios API.

CONTROL DE VERSIONES

- Los commits siguen el estándar de mensajes y estructura definido en 'Buenas Prácticas para escribir commits' versión 0.0.1.

HISTORIAS DE USUARIO

- Las historias de usuario incluyen criterios de aceptación y estos son verificados por el equipo que se cumplen.

PRUEBAS

- El código ha pasado por pruebas unitarias para front y back, con un 100% de éxito en la ejecución de las pruebas.

DESPLIEGUE

- Todos los pipelines automatizados en Azure DevOps para CI/CD están configurados y ejecutan despliegues sin errores a entornos de prueba y producción.
- El sistema de producción se encuentra disponible.

DOCUMENTACION

- Se han actualizado los manuales de usuario para reflejar los nuevos cambios.

Figura 38. Definición de terminado - Versión 2.0

En la **Revisión** del Producto se presentó el incremento de software (Ver Figura 39) cuyo valor hacia el usuario estaba centrado en la adición de los campos necesarios para la gestión de tareas tales como la descripción, la fecha de inicio y fin y responsable delegado a la tarea.

Crear Tarea Cancelar Crear

Nombre

Fecha de Inicio
 📅

Descripción

Fecha de Fin
 📅

Responsable
 ▾

Figura 39. Incremento del Sprint 9

Durante la **Retrospectiva** del Sprint 9, se consideró importante continuar con la refinación en conjunto de las historias de usuario utilizando pizarras colaborativas y Azure Board. Sin embargo, se identificaron problemas que debían ser abordados. Entre ellos, la necesidad de mejorar la precisión en las estimaciones y la comunicación entre los equipos de backend y frontend. También se señaló la importancia de no subestimar las historias de usuario y de evitar detener el Daily por cortes de luz. En la fase de adaptación, se propuso encontrar una estrategia para el Daily asincrónico, la cual era utilizar la aplicación web de Miro para reflejar los avances cuando alguien no pueda asistir al Daily sincrónico. Otra medida elegida fue añadir una sección en la herramienta “Postman” que refleje la trazabilidad de los endpoints desarrollados para poder evaluar el impacto de un cambio en el sistema y medir el nivel de dificultad.

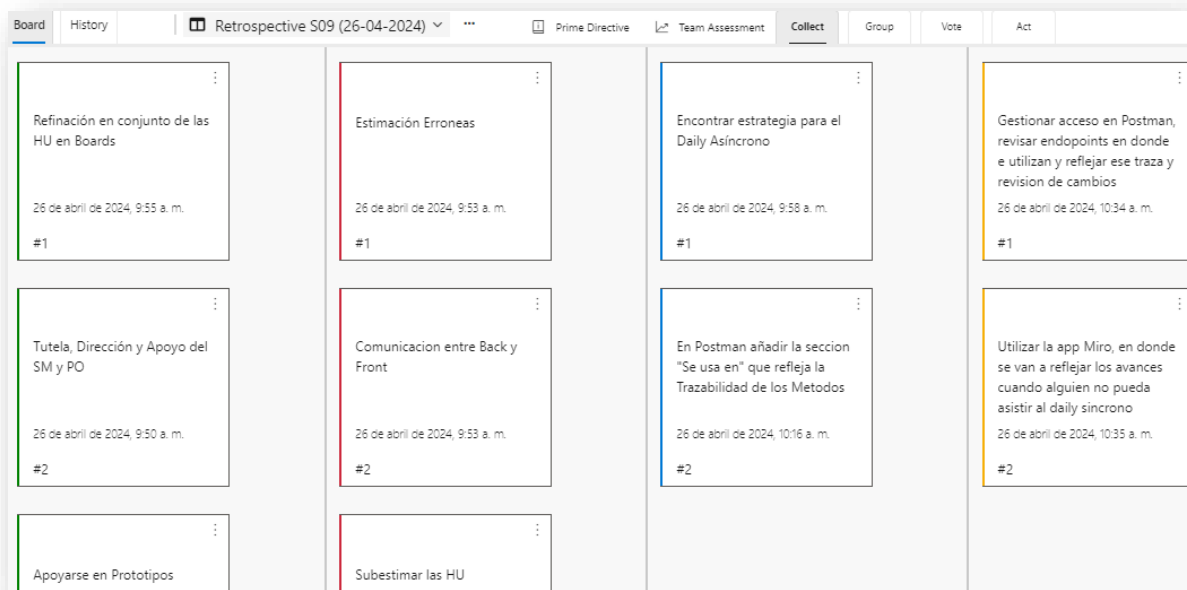


Figura 40. Retrospectiva del Sprint 9

2.6.10 Sprint 10: Visualizar el flujo de tareas del Gestor Empresarial

El **objetivo** del Sprint 10 se centró en visualizar el flujo de tareas del gestor empresarial desde la creación de una tarea a nivel macro hasta el inicio de su ejecución. La **velocidad** del equipo se mantuvo constante en 21 puntos historia, registro basado en la experiencia de sprints anteriores.

Las **historias de usuario** seleccionadas para este sprint incluyeron: la asignación de fechas (HU-11), correspondiente al trabajo pendiente del anterior sprint; la gestión de

proyectos (HU-12), que otorga al gestor empresarial la capacidad de crear, editar y eliminar los proyectos de la empresa; los filtros de proyectos (HU-13) que permiten al gestor buscar los proyectos por nombre y ordenarlos ascendente y descendente según sus características; los filtros de tareas (HU-14), que permiten al gestor empresarial filtrar las tareas por nombre, responsable y estado, y ordenarlas ascendente y descendente según sus características. Entre las historias técnicas, se trabajó en la integración de una nueva biblioteca de tablas en Next.js con el objetivo de facilitar el trabajo a los Scrum developers al reutilizar funciones para el ordenamiento de datos.

Order	Title	Assigned To	State	Priority	Story Points
1	HU-11 Asignación de fechas	GLENDER ISRA...	Closed	2	3
2	HU-14 Filtros de tareas GE	CHRISTIAN PAT...	Closed	1	5
3	HU-12 Gestión de proyectos	CHRISTIAN PAT...	Closed	1	5
4	Librería de tablas	DAVID SEBASTI...	Closed	1	8
5	HU-13 Filtros de proyectos GE	DAVID SEBASTI...	Closed	2	5

Figura 41. Sprint Backlog del Sprint 10

En la Figura 42 se muestra parte del incremento de software presentado en la **Revisión** del Producto del Sprint 10, en donde se generó valor hacia el usuario implementando el flujo de tareas para el Gestor Empresarial a través de las interfaces del Backlog y Tablero.

Nombre ↑↓	Responsable ↑↓	Estado ↑↓	Fecha de inicio ↑↓	Fecha de fin ↑↓	Acciones
Establecer un plan de trabajo	Juan Guerrero	New	2024-07-14	2024-07-19	✍️ 🗑️ ⚙️
Identificar y contratar proveedores	María García	New	2024-07-20	2024-08-10	✍️ 🗑️ ⚙️
Realizar un inventario	María García	New	2024-07-12	2024-08-03	✍️ 🗑️ ⚙️
Clasificar las reparaciones	Juan Guerrero	New	2024-07-19	2024-07-25	✍️ 🗑️ ⚙️
Realizar una evaluación	Glender Miranda	New	2024-07-21	2024-07-28	✍️ 🗑️ ⚙️
Establecer un presupuesto	David Pazmiño	New	2024-07-27	2024-08-04	✍️ 🗑️ ⚙️

Figura 42. Incremento del Sprint 10

Durante la **Retrospectiva** del Sprint 10, se consideró importante continuar con la buena comunicación dentro del equipo. Sin embargo, se identificaron algunos problemas, entre ellos, la necesidad de evitar discutir aspectos técnicos con el PO/Stakeholders fuera de los y realizar la planificación a mitad del sprint. También se resaltó la importancia de no negociar con el PO y de no comprometerse a completar las horas planificadas. En la fase de adaptación, se propuso utilizar la extensión de Planificación “Estimate” para estimar y definir qué historias de usuario se van a trabajar en el sprint. Otra proposición fue dedicar tiempo suficiente a la planificación, priorizar adecuadamente las tareas y organizar el tiempo disponible. También se decidió crear y definir estimaciones y prioridades claras, comprender mejor las historias de usuario y organizar de manera más eficiente el tiempo disponible.

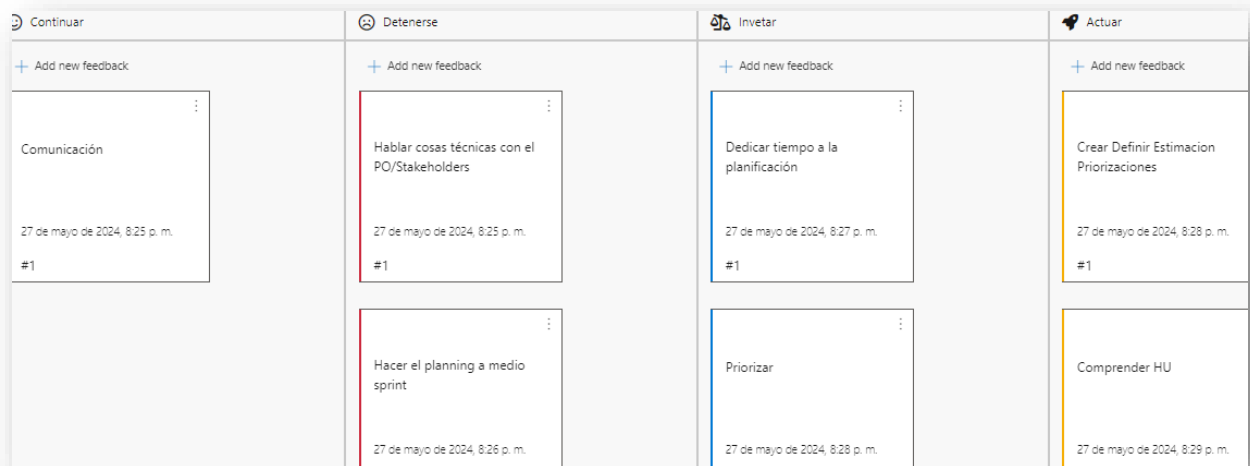


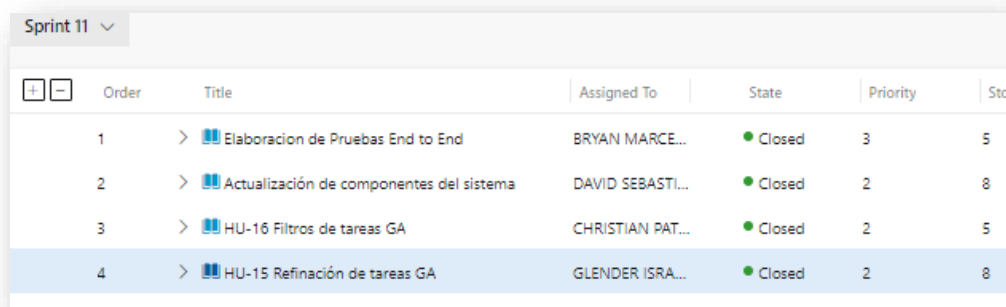
Figura 43. Retrospectiva del Sprint 10

2.6.11 Sprint 11: Gestionar las tareas del Gestor de Área

El **objetivo** del Sprint 11 se centró en gestionar las tareas del Gestor de Área para organizar el trabajo. La **velocidad** del equipo se mantuvo constante en 21 puntos historia, basada en la experiencia de sprints anteriores.

Las **historias de usuario** seleccionadas para este sprint incluyeron: la implementación de filtros de tareas para el GA (HU-16), permitiendo filtrar las tareas por nombre, estado y proyecto, y ordenarlas ascendente y descendentemente para identificar rápidamente tareas específicas entre todas las disponibles; la refinación de tareas (HU-15) para descomponerlas en subtareas más específicas y manejables, esto incluyó la visualización, creación, edición y eliminación de subtareas, así como el inicio del flujo de subtareas. Por otro lado, las historias técnicas que se tomaron en cuenta fueron: la elaboración de pruebas

End to End para mejorar la calidad del producto software; y la actualización de componentes del sistema en base a nuevos prototipos que surgieron a partir de la validación con los stakeholders.



Order	Title	Assigned To	State	Priority	Story Points
1	> Elaboracion de Pruebas End to End	BRYAN MARCE...	Closed	3	5
2	> Actualización de componentes del sistema	DAVID SEBASTI...	Closed	2	8
3	> HU-16 Filtros de tareas GA	CHRISTIAN PAT...	Closed	2	5
4	> HU-15 Refinación de tareas GA	GLENDER ISRA...	Closed	2	8

Figura 44. Sprint Backlog del Sprint 11

Para la estimación de este Sprint 11 se utilizó el plugin de Planificación "Estimate", el cual permitió cuantificar la dificultad de los elementos de trabajo del Sprint Backlog directamente en la plataforma de Azure DevOps. Este plugin permite a los miembros del equipo votar de manera anónima sobre la complejidad de cada historia de usuario utilizando la técnica de Planning Poker. Los votos se recogen y se promedian para obtener una estimación consensuada. Además, "Estimate" permite que las estimaciones se realicen sobre los elementos de trabajo creados en el Product Backlog, estableciendo una trazabilidad directa y precisa con esos elementos, lo que mejora la coherencia y precisión en la planificación del sprint. En la Figura 45 se muestra un ejemplo del uso de este plugin, destacando su utilidad para mejorar la colaboración del equipo y la eficiencia en la estimación de las tareas.

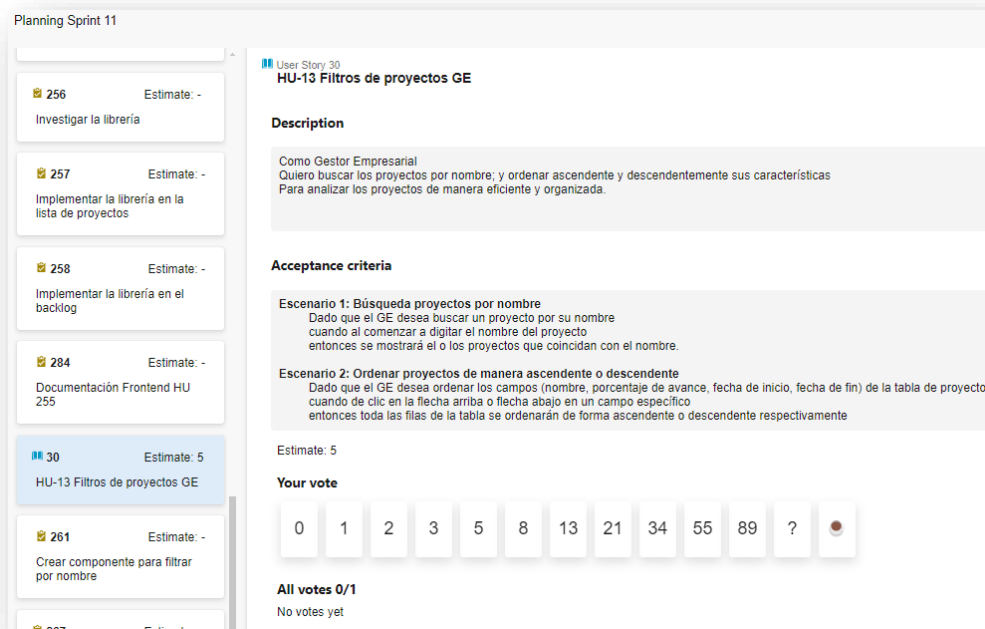


Figura 45. Uso del Plugin 'Estimate' para la Planificación del Sprint 11

Durante las **Daily Scrum**, el equipo no enfrentó inconvenientes significativos y su desarrollo fue fluido y productivo. La buena comunicación y la práctica del Pair Programming dentro del equipo contribuyeron a la resolución efectiva de problemas y a la colaboración constante. En la **Revisión** del Producto, se presentó el valor del incremento hacia el usuario a través de la refinación de tareas más complejas a tareas más sencillas, esto utilizando la pantalla del Backlog del Gestor de Área (Ver Figura 46).

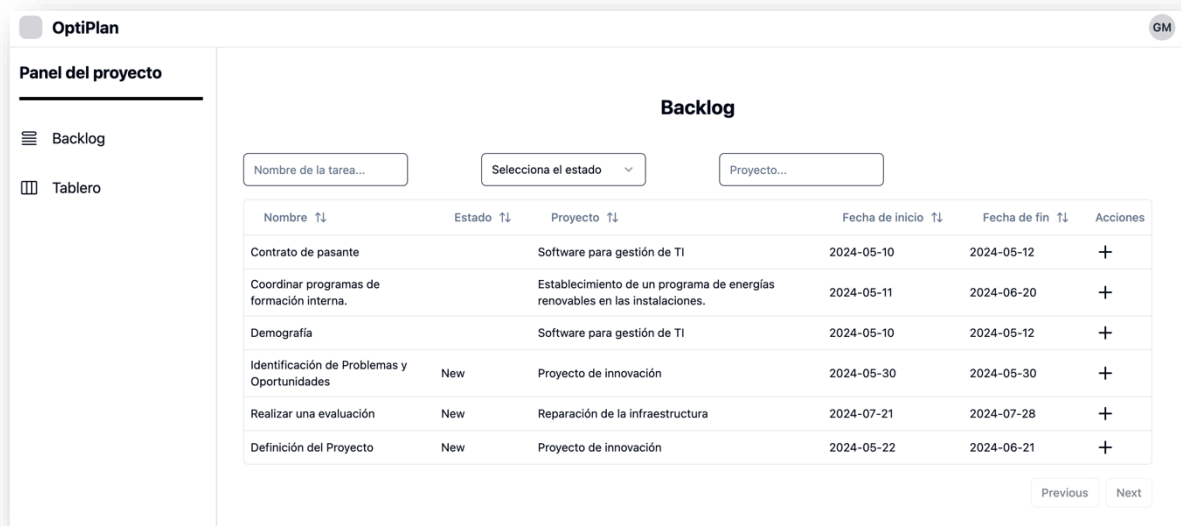


Figura 46. Incremento del Sprint 11

Durante la **Retrospectiva** del Sprint 11, se consideró importante continuar con la buena comunicación dentro del equipo y la práctica del pair programming. Sin embargo, se identificaron varios problemas que debían ser abordados, uno de ellos era el comprometerse a cumplir lo que realmente no se va a alcanzar durante el sprint, otro era la tendencia a procrastinar en requisitos que generen mayor valor. También, se señaló la importancia de no descuidar los valores del coraje y compromiso en el proceso de construcción del producto software. En la fase de adaptación, se propuso priorizar adecuadamente las tareas, estableciendo una lista de prioridades para identificar las necesidades más urgentes del usuario. Además, se decidió establecer una medida drástica que contemplaba definir un horario fijo durante la noche destinado únicamente para construir el producto software en pareja. Finalmente, se recomendó utilizar la técnica Pomodoro para mejorar la concentración y productividad.

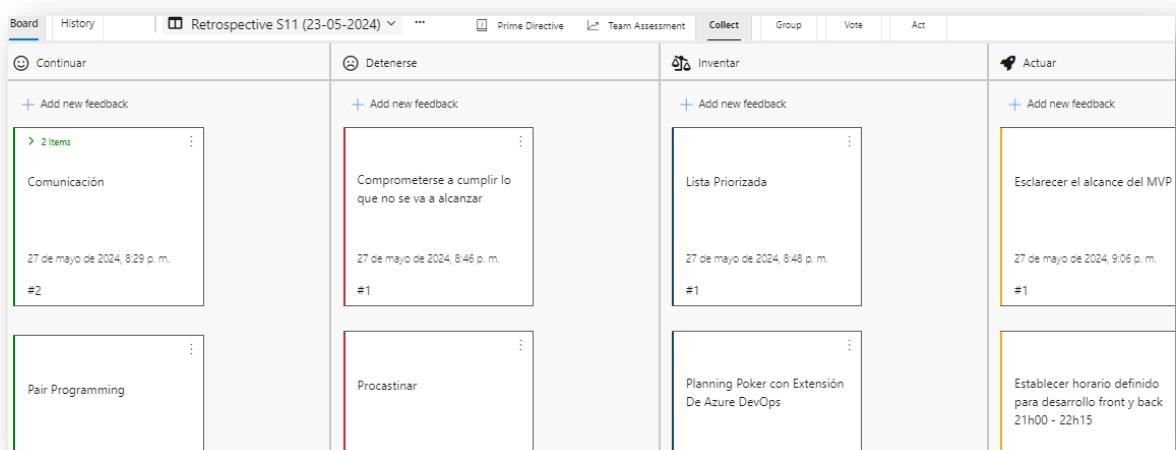


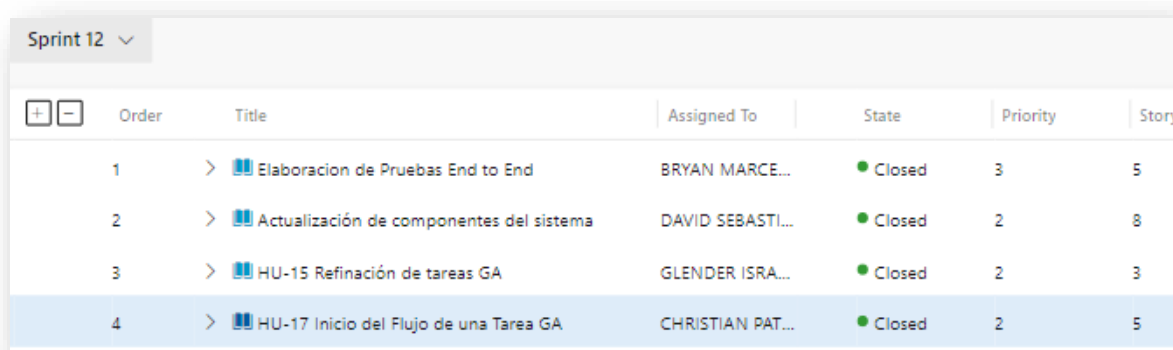
Figura 47. Retrospectiva del Sprint 11

2.6.12 Sprint 12: Visualizar el flujo de tareas del Gestor de Área

El **objetivo** del Sprint 12 estuvo relacionado con visualizar el flujo de tareas del perfil del gestor de área desde que recibe una tarea macro, la refina en tareas más pequeñas, asigna personal e inicia su ejecución. La **velocidad** de equipo se estableció en 21 puntos de historia, siendo el número de puntos de historia que se alcanzaron a completar en el anterior sprint.

Las **historias de usuario** seleccionadas para este Sprint 12 incluyeron: la refinación de tareas (HU-15), permitiendo al GA descomponer las tareas asignadas en subtareas más específicas y manejables; el inicio del flujo de una tarea (HU-17), permitiendo al gestor de

área iniciar el flujo de trabajo de una tarea asignada a un líder de área para que este pudiera visualizarla y comenzar a trabajar en ella. Respecto a historias técnicas, el equipo seleccionó la elaboración de pruebas End to End cuyo objetivo era implementar y automatizar las pruebas para asegurar que los endpoints retornaran los datos requeridos por el frontend. Otra historia técnica fue la actualización de componentes del sistema, incluyendo el Sidebar, formularios, página de información y el tablero Kanban, basándose en los prototipos actualizados. Estas dos historias técnicas se trasladaron nuevamente a este sprint debido a que los Scrum Developers se habían auto organizado para trabajarlas en dos sprints.



Order	Title	Assigned To	State	Priority	Story
1	> Elaboracion de Pruebas End to End	BRYAN MARCE...	Closed	3	5
2	> Actualización de componentes del sistema	DAVID SEBASTI...	Closed	2	8
3	> HU-15 Refinación de tareas GA	GLENDER ISRA...	Closed	2	3
4	> HU-17 Inicio del Flujo de una Tarea GA	CHRISTIAN PAT...	Closed	2	5

Figura 48. Sprint Backlog del Sprint 12

Durante las **Daily Scrum**, el equipo no enfrentó inconvenientes significativos y su desarrollo fue fluido y productivo. La buena comunicación y la práctica del pair programming dentro del equipo contribuyeron a la resolución efectiva de problemas y a la colaboración constante.

En la **Revisión** del Producto, se generó valor hacia el usuario final a través de la funcionalidad de inicio del flujo de tareas en el perfil del Gestor de Área. En la Figura 49 se muestra la pantalla de confirmación para inicio del flujo. Posterior a la inspección, el Product Owner sugirió revisar el tipo de lenguaje que se estaba utilizando en la aplicación.

¿Estás seguro de que deseas iniciar el flujo de esta tarea?

Proyecto: Rediseño de la malla curricular

Tarea Padre: Identificación de Problemas y Oportunidades

Nombre: Rediseño de la malla curricular

Figura 49. Incremento del Sprint 12

Durante la **Retrospectiva** del Sprint 12, se consideró importante continuar con la práctica del pair programming posterior a la inspección de las Dailys Scrum y seguir fomentando la comunicación por todos los medios disponibles. No obstante, se identificaron prácticas que debían detenerse, como no agregar al sprint backlog de Azure DevOps las historias de usuario técnicas relacionadas con los prototipos, ya que esto impedía reflejar el esfuerzo real de los desarrolladores. También se volvió hacer hincapié en la mala práctica de comprometerse a cumplir lo que realmente no se va a alcanzar durante el sprint. En la fase de adaptación, se recomendaron varias acciones para mitigar los problemas identificados. Se propuso la recomendación de dedicarse exclusivamente a verificar el cumplimiento de la Definición de Terminado previo a la revisión del producto. Un compromiso como equipo fue dividir las tareas y hacerse cargo de ellas de manera rigurosa para mejorar la organización del trabajo.

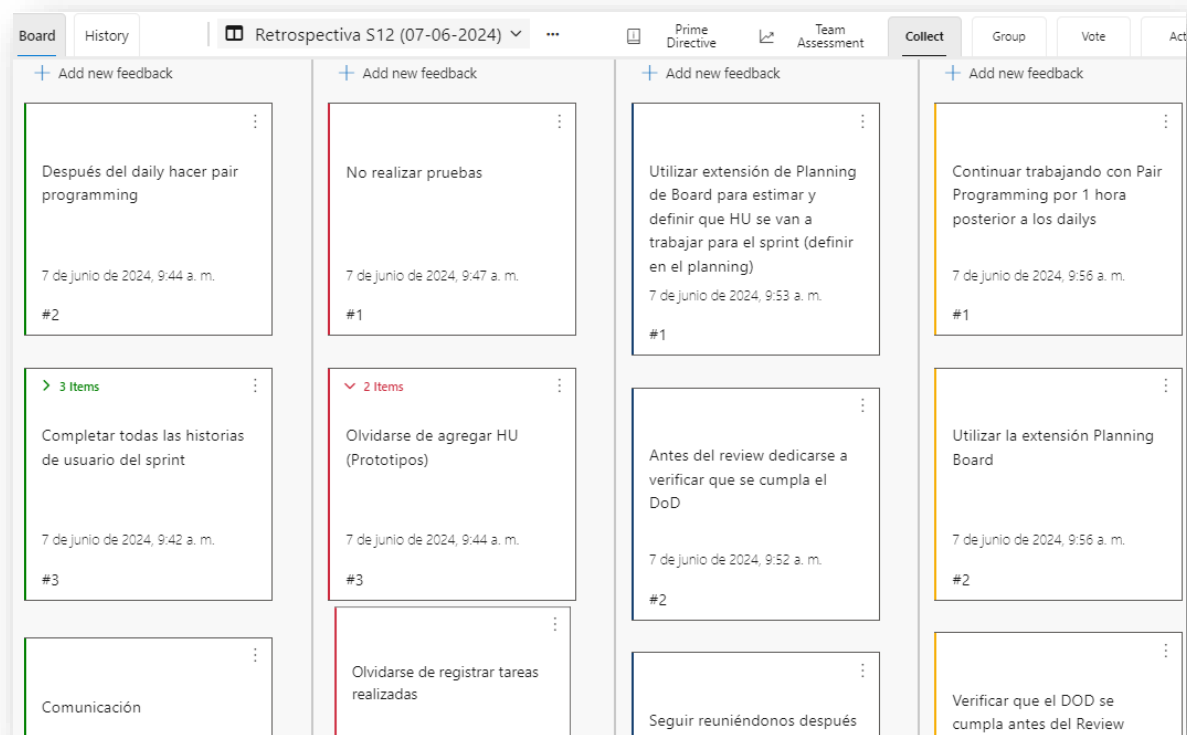


Figura 50. Retrospectiva del Sprint 12

2.6.13 Sprint 13: Gestionar las tareas del Líder de Área

El **objetivo del sprint 13** estuvo relacionado a la gestión de tareas y subtareas del perfil Líder de área. La **velocidad** del equipo se mantuvo en 21 puntos de historia.

Las **historias de usuario** seleccionadas para el Sprint 13 fueron: la HU-18 relacionada con la refinación de tareas del Líder de Área, cuyo objetivo era descomponer las tareas asignadas en subtareas más específicas y manejables; la HU-19 relacionada con los filtros de tareas que permite al Líder de Área buscar las tareas por nombre, estado y proyecto, y ordenar ascendente y descendente las tareas; la HU-20 abarca el inicio del flujo de trabajo de una tarea facilitando al Líder de Área el seguimiento del progreso de la tarea. Respecto a historias técnicas, se encontraban presentes el descubrimiento del producto para el líder de área, la cual buscaba definir los prototipos de dicho perfil. La otra historia técnica era acerca descubrimiento del producto para el perfil del Técnico, la cual buscaba anticiparse al siguiente sprint al definir los prototipos de dicho perfil.

Order	Title	Assigned To	State	Priority	Story Points
1	Descubrimiento del producto LA	DAVID SEBASTI...	Closed	2	3
2	HU-18 Refinación de tareas LA	GLENDER ISRA...	Closed	2	5
3	HU-19 Filtros de tareas LA	CHRISTIAN PAT...	Closed	2	3
4	HU-20 Inicio del Flujo de Tareas LA	BRYAN MARCE...	Closed	2	5
5	Descubrimiento del producto Tecnico	DAVID SEBASTI...	Closed	2	5

Figura 51. Sprint Backlog del Sprint 13

El desarrollo de las **reuniones diarias** fue correcto y satisfactorio, el equipo se mantuvo concentrado en sus tareas y colaboró de manera efectiva, utilizando prácticas ágiles como el pair programming para solucionar conflictos y errores de código de forma eficiente. En la Figura 52 se muestra la funcionalidad del backlog para el perfil de Líder de Área presentada durante la inspección de la **Revisión** del Producto. Los comentarios del Product Owner fueron positivos al visualizar una interfaz limpia y con las funcionalidades justas y necesarias, generando valor al usuario final.

Nombre	Estado	Proyecto	Fecha de inicio	Fecha de fin	Acciones
Mapa investigación	New	Software para gestión de TI	2024-06-24	2024-06-25	+
Gestión de tareas	New	Software para gestión de TI	2024-06-21	2024-06-22	+
Levantamiento de Datos	New	Reparación de la infraestructura	2024-07-25	2024-07-30	+
Evaluación de Estructuras	New	Reparación de la infraestructura	2024-07-28	2024-08-03	+

Figura 52. Incremento del Sprint 13

Durante la **Retrospectiva** del Sprint 13, se destacó la importancia de continuar con el descubrimiento del producto y la práctica del pair programming, reconociendo su impacto positivo en el rendimiento del equipo. También se identificaron problemas clave, como la falta de compromiso para asistir a las reuniones y las demoras en las sesiones de prototipos entre los Scrum Developer y el Product Owner, esto debido a la dificultad de horarios y

duración de las sesiones. Otro problema que se debía abordar fue el consumo excesivo de créditos en las máquinas virtuales donde estaban desplegados los entornos de producción y desarrollo. En la fase de adaptación, se propusieron varias mejoras, por ejemplo: se sugirió establecer días específicos para encender la infraestructura, buscando optimizar el uso de recursos disponibles como estudiantes. Además, se sugirió crear lo más pronto el elemento de trabajo relacionado con la actividad de descubrimiento del producto para el perfil del técnico, con el fin de evitar demoras en el progreso de la construcción del producto software.

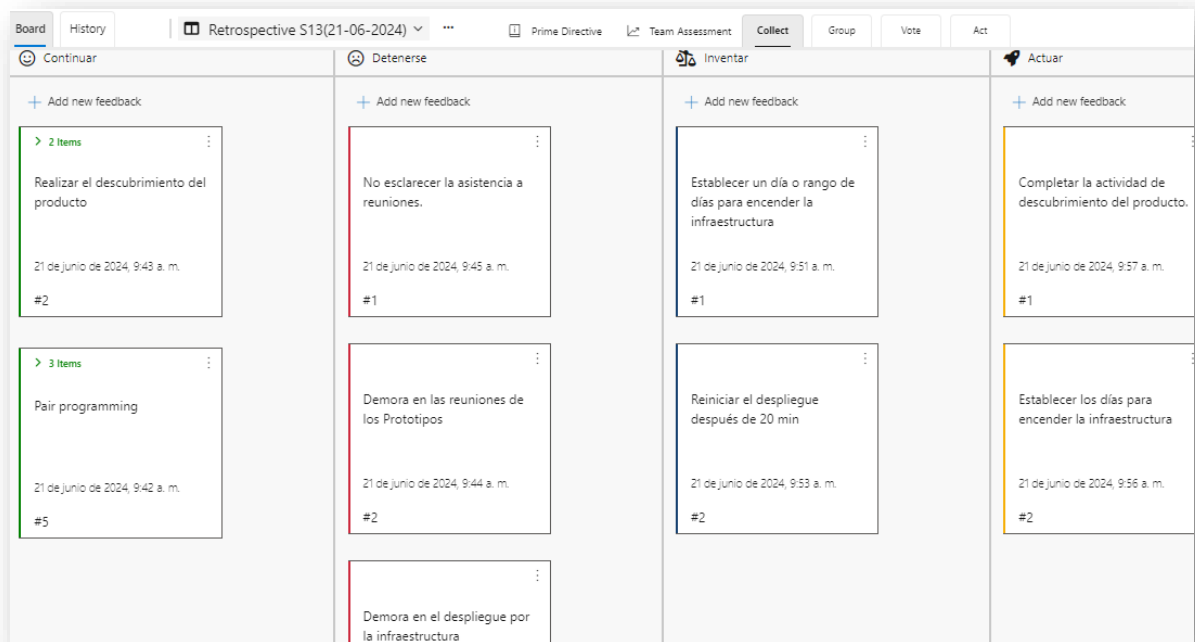


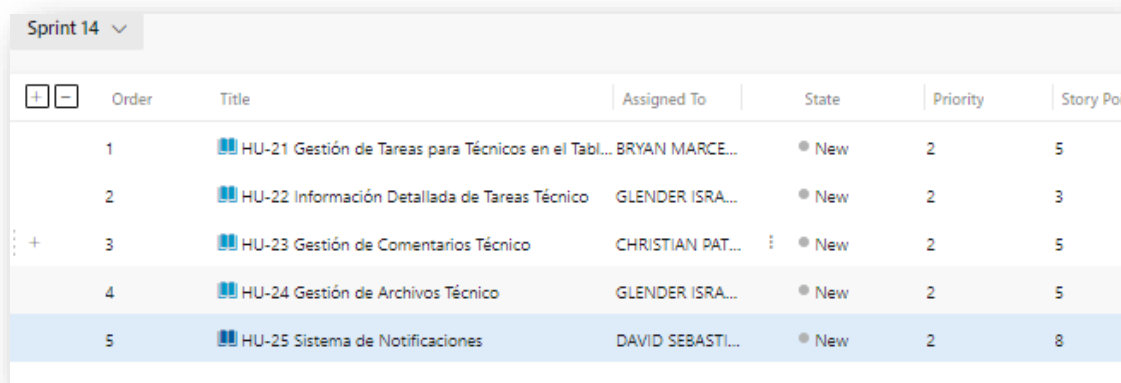
Figura 53. Retrospectiva del Sprint 13

2.6.14 Sprint 14: Gestionar la ejecución de tareas del Técnico

El **objetivo** del sprint 14 estuvo relacionado a la gestión del flujo de tareas operativas del perfil Técnico. La **velocidad** del equipo se estableció en 21 puntos de historia.

Las **historias de usuario** seleccionadas para el Sprint 14 incluyeron la gestión de tareas para Técnicos en el tablero Kanban (HU-21), donde los Técnicos podían arrastrar y soltar sus tareas para moverlas a través de su flujo y gestionar su estado y progreso. También se abordó la necesidad de visualizar información detallada de las tareas en el tablero Kanban (HU-22), proporcionando a los Técnicos una perspectiva completa de su estado, proyecto, fechas y descripción. Además, se implementó la gestión de comentarios y

archivos (HU-23 y HU-24) en las tareas del tablero Kanban para facilitar la comunicación y el seguimiento dentro del equipo. Asimismo, se escogió la historia de usuario relacionada con un sistema de notificaciones (HU-25) para que los roles con subordinados de menor jerarquía reciban alertas cuando se suba un archivo, se agregue un comentario o se cambie el estado de una tarea, permitiendo una supervisión adecuada del progreso de los subordinados.



Order	Title	Assigned To	State	Priority	Story Points
1	HU-21 Gestión de Tareas para Técnicos en el Tabl...	BRYAN MARCE...	New	2	5
2	HU-22 Información Detallada de Tareas Técnico	GLENDER ISRA...	New	2	3
3	HU-23 Gestión de Comentarios Técnico	CHRISTIAN PAT...	New	2	5
4	HU-24 Gestión de Archivos Técnico	GLENDER ISRA...	New	2	5
5	HU-25 Sistema de Notificaciones	DAVID SEBASTI...	New	2	8

Figura 54. Sprint Backlog del Sprint 14

En las **sesiones diarias** de Scrum, el equipo no reporto problemas y se mantuvo enfocado en sus tareas, fomentando el valor del coraje y compromiso.

En la **Revisión** del Producto, los Scrum Developers destacaron la implementación del tablero Kanban, la visualización de información detallada, la gestión de comentarios y archivos, y el sistema de notificaciones. Estas mejoras permitieron otorgar mayor valor hacia el usuario final ya que abarcaban todas las características mínimas liberables que se establecieron en la planificación del Release 2. En la Figura 55 se muestra la funcionalidad de la gestión de flujo de tareas en el tablero Kanban para el perfil del Técnico.

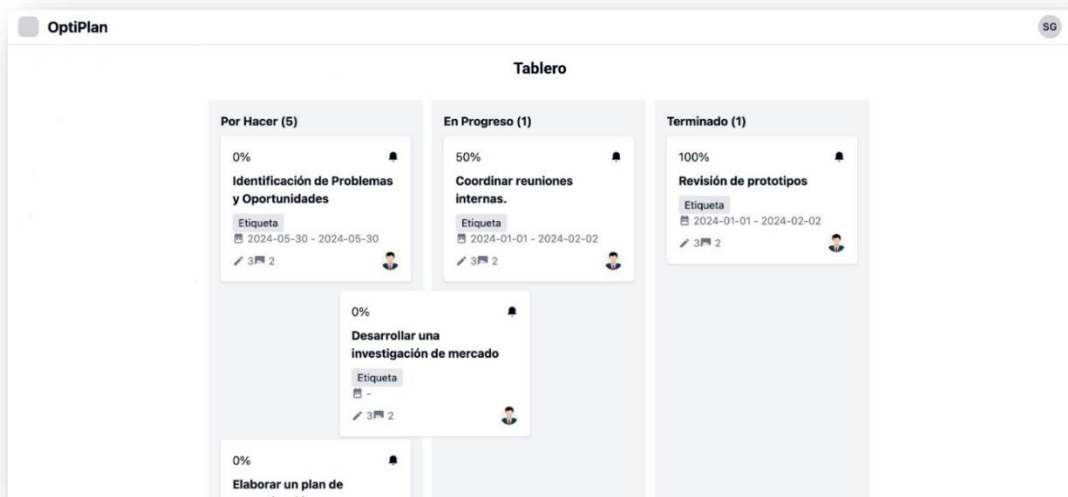


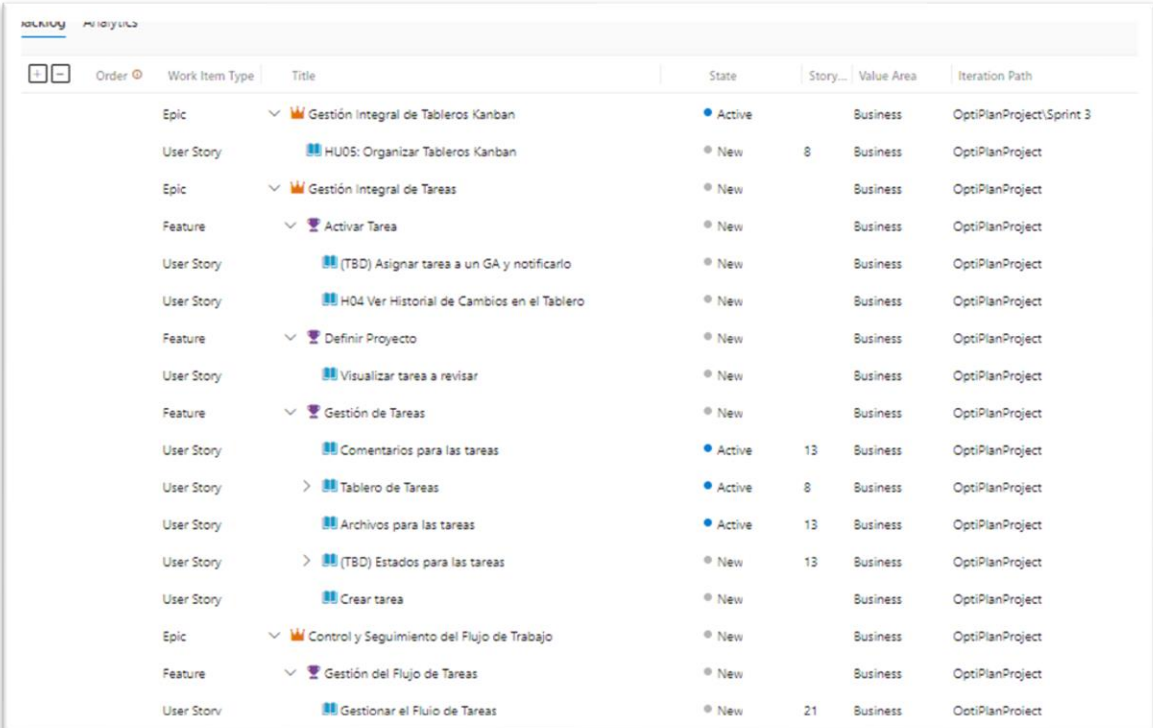
Figura 55. Incremento del Sprint 14

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Gracias a la plataforma Azure DevOps, el equipo de trabajo tuvo acceso constante a información clave, lo que aseguró transparencia a la información crucial, facilitando la actualización y revisión de artefactos esenciales como el Product Backlog, Sprint Backlog, Wiki y los Pipelines de Integración continua. Esta plataforma fue clave en cada Sprint, ayudando al equipo a planificar su trabajo basándose en los objetivos acordados entre el Product Owner y los Scrum Developers. Azure DevOps también gestionó de manera eficiente el Product Backlog, permitiendo crear, actualizar, eliminar, priorizar y estimar elementos de trabajo (historias de usuario).

En la Figura 56 se muestra una parte del Product Backlog de Optiplan en Azure DevOps. Los elementos de trabajo están jerarquizados desde Épicas, Características e Historias de Usuario para una mejor abstracción y comprensión de los requisitos.



Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path
	Epic	👑 Gestión Integral de Tableros Kanban	● Active		Business	OptiPlanProject\Sprint 3
	User Story	👤 HU05: Organizar Tableros Kanban	● New	8	Business	OptiPlanProject
	Epic	👑 Gestión Integral de Tareas	● New		Business	OptiPlanProject
	Feature	👤 Activar Tarea	● New		Business	OptiPlanProject
	User Story	👤 (TBD) Asignar tarea a un GA y notificarlo	● New		Business	OptiPlanProject
	User Story	👤 H04 Ver Historial de Cambios en el Tablero	● New		Business	OptiPlanProject
	Feature	👤 Definir Proyecto	● New		Business	OptiPlanProject
	User Story	👤 Visualizar tarea a revisar	● New		Business	OptiPlanProject
	Feature	👤 Gestión de Tareas	● New		Business	OptiPlanProject
	User Story	👤 Comentarios para las tareas	● Active	13	Business	OptiPlanProject
	User Story	👤 Tablero de Tareas	● Active	8	Business	OptiPlanProject
	User Story	👤 Archivos para las tareas	● Active	13	Business	OptiPlanProject
	User Story	👤 (TBD) Estados para las tareas	● New	13	Business	OptiPlanProject
	User Story	👤 Crear tarea	● New		Business	OptiPlanProject
	Epic	👑 Control y Seguimiento del Flujo de Trabajo	● New		Business	OptiPlanProject
	Feature	👤 Gestión del Flujo de Tareas	● New		Business	OptiPlanProject
	User Story	👤 Gestionar el Flujo de Tareas	● New	21	Business	OptiPlanProject

Figura 56. Product Backlog en Azure DevOps

En la Figura 57 se presenta un ejemplo del Sprint Backlog correspondiente al sprint número 13.

Order	Title	State	Assigned To	Story Points	Priority
1	Descubrimiento del producto LA	Closed	DAVID SEBASTI...	3	2
2	Refinación de tareas LA	Closed	GLENDER ISRA...	5	2
3	Filtros de tareas LA	Closed	CHRISTIAN PAT...	3	2
4	Inicio del Flujo de Tareas LA	Closed	BRYAN MARCE...	5	2
5	Descubrimiento del producto Técnico	Closed	DAVID SEBASTI...	5	2

Figura 57. Sprint Backlog del Sprint 13

Un aspecto importante por destacar es que las historias de usuario elaboradas durante la planificación inicial del release se refinaron y actualizaron durante la ejecución de los sprints. Estas historias se ajustaron y evolucionaron, e incluso se crearon nuevas, conforme avanzaba el descubrimiento del producto y se discutían las necesidades de los usuarios finales. Esta adaptación continua permitió al equipo ajustarse mejor a los requerimientos reales y mejorar la alineación del producto con las expectativas de los usuarios.

Otra parte esencial para gestionar la construcción del producto software fue la sección "Wiki" dentro de Azure DevOps, que permitió documentar y transparentar la información relevante del proyecto de manera estructurada. En la Figura 58 se presenta una captura del índice de toda la documentación manejada por el equipo Scrum.

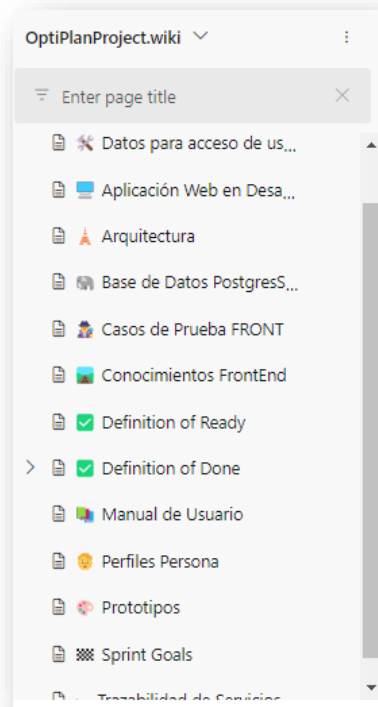


Figura 58. Wiki con información del proyecto

En la Figura 59 se observa el historial de grabaciones almacenadas en la herramienta de Microsoft Teams que permitieron a los desarrolladores realizar el evento del Daily Scrum. En este evento se realiza la inspección para conocer el progreso de construcción del incremento y la adaptación para solucionar los problemas o impedimentos encontrados durante la ejecución del sprint.

Name	Modified	Modified By
Scrum Daily-20240507_203610-Grabación ...	May 7	GLENDER ISRAEL MIRAJ
Scrum Daily-20240509_205144-Grabación ...	May 9	GLENDER ISRAEL MIRAJ
Scrum Daily-20240514_203725-Meeting Re...	May 14	CHRISTIAN PATRICIO F
Scrum Daily-20240520_204650-Meeting Re...	May 20	CHRISTIAN PATRICIO F
Scrum Daily-20240522_204534-Grabación ...	May 22	GLENDER ISRAEL MIRAJ
Scrum Daily-20240527_203814-Meeting Re...	May 27	CHRISTIAN PATRICIO F
Scrum Daily-20240528_203938-Meeting Re...	May 28	CHRISTIAN PATRICIO F
Scrum Daily-20240529_203428-...	May 29	CHRISTIAN PATRICIO F
Scrum Daily-20240531_205601-Meeting Re...	May 31	CHRISTIAN PATRICIO F
Scrum Daily-20240610_204127-Grabación ...	June 10	GLENDER ISRAEL MIRAJ

Figura 59. Registro de grabaciones de las Daily Scrum en Microsoft Teams

El plugin de Planificación "Estimate" permitió realizar estimaciones de historias de usuario de manera efectiva y colaborativa. Sus grandes ventajas incluyen la capacidad de organizar sesiones de planificación en línea, la simplicidad de uso y la facilidad para involucrar a todo el equipo en el proceso de estimación. Además, permite estimar directamente sobre los elementos de trabajo creados en el Product Backlog, estableciendo una trazabilidad directa con esos elementos. En la Figura 60 se muestran las sesiones de planificación utilizando este plugin, destacando su utilidad para mejorar la planificación del sprint.

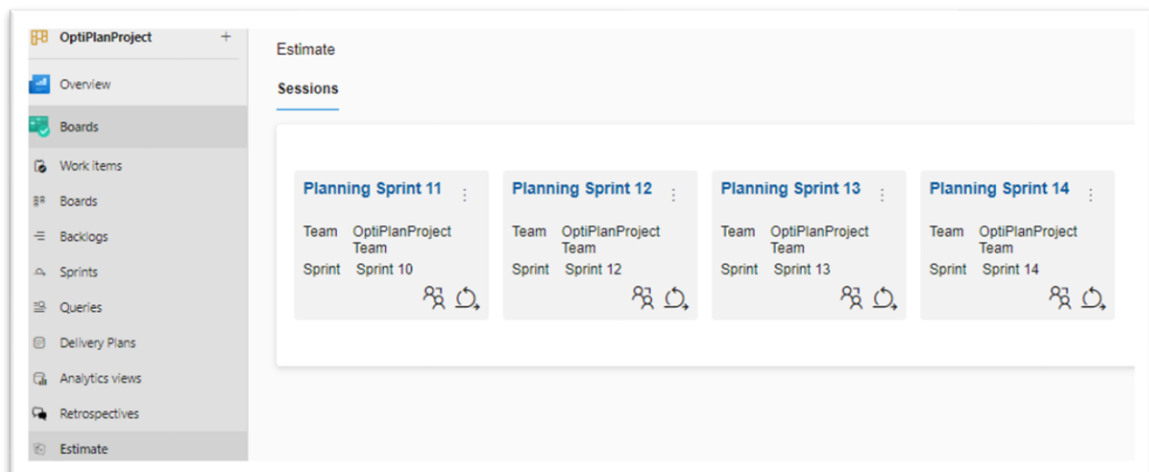


Figura 60. Sesiones de estimación en Azure DevOps

Para las revisiones del Sprint se utilizó material de presentación que permitió a los asistentes visualizar claramente el progreso y los logros alcanzados. Estas presentaciones

generan valor al facilitar la comunicación efectiva y asegurar que todos los miembros del equipo y stakeholders comprendan el estado del proyecto. En la Figura 61 se incluye una captura de la presentación utilizada en la Revisión del Sprint 13, esta muestra la comparación entre las historias de usuario planificadas y las completadas.



Figura 61. Diapositiva utilizada en la Revisión del Sprint 13

El plugin de Retrospectivas de Azure DevOps permitió al equipo registrar sus observaciones y propuestas de una manera estructurada y visualmente accesible. Gracias a esta herramienta, se pudieron identificar claramente los puntos fuertes, las áreas de mejora y las acciones concretas a implementar, facilitando así la inspección y adaptación del proceso del Marco Scrum. En la Figura 62 se listan una parte de todas las retrospectivas realizadas por el equipo de desarrollo, proporcionando información como el nombre de cada retrospectiva, la fecha en que se creó, la cantidad de elementos de trabajo pendientes y el total de elementos de trabajo detallados en cada sesión.

Retrospective Name	Created Date	Pending Work Items	Total Work Items
Retrospective S01 (29-11-23)	Jul 17, 2024	0	1
Retrospective S02 (14-12-23)	Jul 24, 2024	0	0
Retrospective S04 (18-01-24)	Jan 18, 2024	0	3
Retrospective S03 (04-01-24)	Jul 24, 2024	0	0
Retrospective S05 (1-2-2024)	Feb 1, 2024	0	0
Retrospective S06 (15-2-2024)	Feb 15, 2024	0	2
Retrospective S07 (29-2-2024)	Jul 24, 2024	1	1
Retrospective S08 (11-04-24)	Apr 12, 2024	0	2

Figura 62. Historial de Retrospectivas en Azure DevOps

3.2 Conclusiones

Los enfoques tradicionales de desarrollo, que dependen de planes detallados y suponen que los requisitos no cambiarán, suelen fracasar en entornos complejos y adaptativos. En contraste, Scrum reconoce que no es posible predecir todo desde el inicio y centra sus esfuerzos en el empirismo y en los principios Lean. Por un lado, el empirismo permite construir y adaptar el producto software a medida que se aprende sobre el producto. Por otro lado, los principios Lean permiten evitar el desperdicio y trabajar en los artefactos justo cuando son necesarios y en la cantidad suficiente.

Scrum permite realizar la planificación de un sprint mediante un objetivo comercial que genera valor al usuario final. A partir de este objetivo se selecciona requisitos de usuario que permiten crear un incremento de software que satisface a dicho objetivo. Por lo tanto, el propósito de un sprint es validar que el incremento funcional satisface los requisitos del usuario y genera valor comercial.

En la ejecución del sprint, es decir, en la construcción del incremento del producto software se realizan las Daily Scrum para conocer el proceso de la construcción del incremento. En este evento, la inspección consiste en compartir el trabajo de cada desarrollador y los problemas o impedimentos que no le permiten avanzar. Entonces, la adaptación permite solucionar los problemas o impedimentos para finalizar la construcción del incremento.

La Definición de Terminado (DoD, del inglés Definition of Done) es una lista de verificación que permite decidir si el incremento del producto software es potencialmente entregable. Además, el DoD permite garantizar la calidad del producto software que se entregará al usuario. Una vez que el incremento cumple con los criterios de calidad (es decir, el DoD) se realiza la revisión del sprint, que consiste en la inspección y adaptación del producto. Por un lado, la inspección del producto permite mediante una demostración explicar la funcionalidad del incremento de software y pasar a un debate o discusión sobre dicho incremento. Por otro lado, la adaptación del producto permite crear nuevos requisitos de usuario que serán consideradas en próximos sprints.

Finalmente, la retrospectiva del sprint nos permite inspeccionar y adaptar el proceso implementado del marco Scrum. Por un lado, la inspección nos permite identificar qué aspectos se realizaron correctamente y deben continuarse, así como aquellos que presentaron problemas y deben ser detenidos. Con la información subjetiva sobre el proceso se obtiene el contexto para decidir que se desea hacer de manera diferente. Por otro lado, la adaptación consiste en decidir cómo actuar para mejorar el proceso, que puede resultar en historias de usuario o compromisos del equipo de Scrum.

El objetivo de generar valor al usuario final mediante la entrega continua de incrementos funcionales de software fue una prioridad central a lo largo del proyecto. Utilizando la metodología Scrum y herramientas como Azure DevOps, el equipo gestionó la entrega de incrementos de manera adecuada. Durante cada sprint, se entregaron incrementos funcionales que abordaron necesidades específicas del usuario final. La retroalimentación constante del usuario final fue fundamental en este proceso, permitiendo al equipo alinear el producto con las expectativas y necesidades reales de los usuarios. No obstante, se enfrentaron desafíos importantes tales como la aparición de cambios en los requisitos y la complejidad técnica de algunas funcionalidades. Estas dificultades fueron superadas aplicando los valores de Scrum y principios ágiles, que guiaron la colaboración y la resolución de problemas dentro del equipo y con los stakeholders. En conclusión, el enfoque ágil de Scrum y las herramientas utilizadas permitieron un desarrollo adaptativo y centrado en el usuario, resultando en un producto que no solo cumple con los requisitos técnicos, sino que también proporciona un valor significativo y tangible a sus usuarios finales.

3.3 Recomendaciones

Se recomienda que durante la reunión diaria (Daily Scrum) se eviten interrupciones para asegurar que cada miembro tenga tiempo suficiente para responder. Para temas que requieren más tiempo, se sugiere utilizar el método de "estacionamiento virtual", anotando los asuntos pendientes en una pizarra y gestionándolos al final de la reunión.

Se recomienda utilizar favorablemente los principios INVEST para la redacción de historias de usuario ya que su simplicidad en su estructura facilita la comunicación y la colaboración entre las diferentes partes interesadas.

Se recomienda refinar continuamente la Definición de Terminado para asegurar que cada incremento de software cumpla con los estándares de calidad esperados y esté alineado con los objetivos del proyecto

Para una gestión efectiva del backlog en Azure DevOps, se recomienda, al trasladar una historia de usuario incompleta de un sprint al siguiente, crear una copia de la historia original sin incluir sus elementos hijos. Esta práctica garantiza que las tareas hijas resueltas en cada sprint se registren correctamente como actividades distintas y específicas a su respectivo sprint, manteniendo así la trazabilidad de dichas tareas.

Se recomienda no crear historias de usuario posterior a la planificación del sprint, únicamente agregar nuevas tareas dentro de las historias de usuario existentes, para mantener la coherencia y el enfoque del equipo en los objetivos establecidos. Esto asegura

que el alcance del sprint no se expanda de manera incontrolada, evitando sobrecargar al equipo y garantizando que las prioridades se gestionen adecuadamente.

4 REFERENCIAS BIBLIOGRÁFICAS

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. VTT Publications, 478, 3–107. <https://arxiv.org/abs/1709.08439v1>
- Atlassian. (n.d.). Los tres pilares del scrum: conoce los principios fundamentales del scrum. 2023. Retrieved June 17, 2024, from <https://www.atlassian.com/es/agile/project-management/3-pillars-scrum>
- Bermejo, M. (2011). El Kanban. Barcelona, España: UOC, 8.
- Braude, E. J., & Bernstein, M. E. (2016). Software Engineering: Modern Approaches, Second Edition. John Wiley. <https://books.google.com.ec/books?id=kLICwAAQBAJ>
- Castellano Lendínez, L. (2019). Kanban. Metodología para aumentar la eficiencia de los procesos. 3C Tecnología. Glosas de innovación aplicadas a la pyme. 8(1), 30–41. <https://doi.org/10.17993/3ctecno/2019>
- chcomley, Jackson, S., & Thurman, P. (2024). Procesos predeterminados y plantillas de procesos - Azure Boards | Microsoft Learn. <https://learn.microsoft.com/es-es/azure/devops/boards/work-items/guidance/choose-process?view=azure-devops&tabs=agile-process>
- Dave West. (n.d.). Sprint Planning | Atlassian. 2020. Retrieved June 24, 2024, from <https://www.atlassian.com/agile/scrum/sprint-planning>
- Default processes and process templates - Azure Boards | Microsoft Learn. (2024). <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/choose-process?view=azure-devops&tabs=agile-process>
- ESIC Business & Marketing School. (2023). Conoce los 12 principios del Manifiesto Agile | ESIC. <https://www.esic.edu/rethink/tecnologia/manifiesto-agile-valores-12-principios-c>
- Geoffrey A. Moore. (1999). Crossing_the_Chasm (Regis McKenna, Ed.). HarperBusiness.
- Himanshu, P. (2024). Getting Started with Azure DevOps Boards and Repos / Blogs / Perficient. <https://blogs.perficient.com/2024/05/15/getting-started-with-azure-devops-boards-and-repos/>
- Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, & Dave Thomas. (2001). Manifiesto por el Desarrollo Ágil de Software. <https://agilemanifesto.org/iso/es/manifesto.html>
- Microsoft. (2024). Azure Repos documentation - Azure DevOps | Microsoft Learn. <https://learn.microsoft.com/en-us/azure/devops/repos/?view=azure-devops>
- Pavel Naydenov. (2022). ¿Qué es Kanban? Principales características y funciones. <https://businessmap.io/es/recursos-de-kanban/primeros-pasos/que-es-kanban>

- ¿Qué es Azure DevOps? - Azure DevOps | Microsoft Learn. (2024).
<https://learn.microsoft.com/es-es/azure/devops/user-guide/what-is-azure-devops?view=azure-devops#choose-azure-devops-services>
- Rea, D., Ovhal, P., & KathrynEE. (2024). Agile workflow in Azure Boards - Azure Boards | Microsoft Learn. <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process-workflow?view=azure-devops>
- Real Academia Española. (2024a). ágil | Diccionario del estudiante | RAE.
<https://www.rae.es/diccionario-estudiante/%C3%A1gil>
- Real Academia Española. (2024b). agilidad | Definición | Diccionario de la lengua española | RAE - ASALE. <https://dle.rae.es/agilidad>
- Rohm William Anton, Jackson, S., & chcomley. (2024). Best practices for Agile project management - Azure Boards | Microsoft Learn. <https://learn.microsoft.com/en-us/azure/devops/boards/best-practices-agile-project-management?view=azure-devops&tabs=agile-process>
- Rubin, K. S. (2012). Essential Scrum : a practical guide to the most popular agile process. Pearson Education.
- Ruiz, J. (2018). Los 4 Valores de la Agilidad: El Manifiesto Ágil - Jorge Ruiz Agile.
<https://jorgeruizagile.com/2018/08/04/los-4-valores-de-la-agilidad-el-manifiesto-agil/>
- Schwaber, K., & Sutherland, J. (2020). La Guía Definitiva de Scrum: Las Reglas Del Juego. Scrum.org. <https://books.google.com.ec/books?id=N9kkzgEACAAJ>
- Sentrio. (2021). Metodologías Agile: los 4 valores y 12 principios del Manifiesto Ágil.
<https://sentrio.io/blog/valores-principios-agile-manifiesto-agil/>
- Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. Harvard Business Review, 85(11), 68.
- Uribe, E. H., & Ayala, L. E. V. (2007). Del manifiesto ágil sus valores y principios. Scientia et Technica, 13(34), 381–386.

5 ANEXOS

5.1 ANEXO I

ANEXO I. Captura de los objetivos por sprint en la Wiki de Azure DevOps

Sprint Goals

 JULIO CESAR SANDOBALIN GUA... 28 jun

- Sprint 1: Crear tableros Kanban (General).
- Sprint 2: Crear flujos de tareas en Kanban (General).
- Sprint 3: Controlar el acceso a los tableros Kanban (General/Seguridad).
- Sprint 4: Mejorar el flujo de trabajo de los tableros Kanban (General).
- Sprint 5: Crear proyectos para las tareas operativas (GE).
- Sprint 6: Mejorar los flujos de trabajo de las tareas operativas (General).
- Sprint 7: Visualizar proyectos y tareas operativas (GE).
- Sprint 8: Descubrir el flujo de tareas para todos los roles (TODOS).
- Sprint 9: Gestionar las tareas del Gestor Empresarial (GE).
- Sprint 10: Visualizar el flujo de tareas del Gestor Empresarial (GE).
- Sprint 11: Gestionar las tareas del Gestor de Área (GA).
- Sprint 12: Visualizar el flujo de tareas del Gestor de Área (GA).
- Sprint 13: Gestionar las tareas del Líder de Área (LA).
- Sprint 14: Gestionar la ejecución de tareas del Técnico (Técnico).

5.2 ANEXO II

ANEXO II. Enlace a la Pizarra Colaborativa de Miro para las Dailys Scrum

https://miro.com/app/board/uXjvNLxy|ps=?share_link_id=237372294523