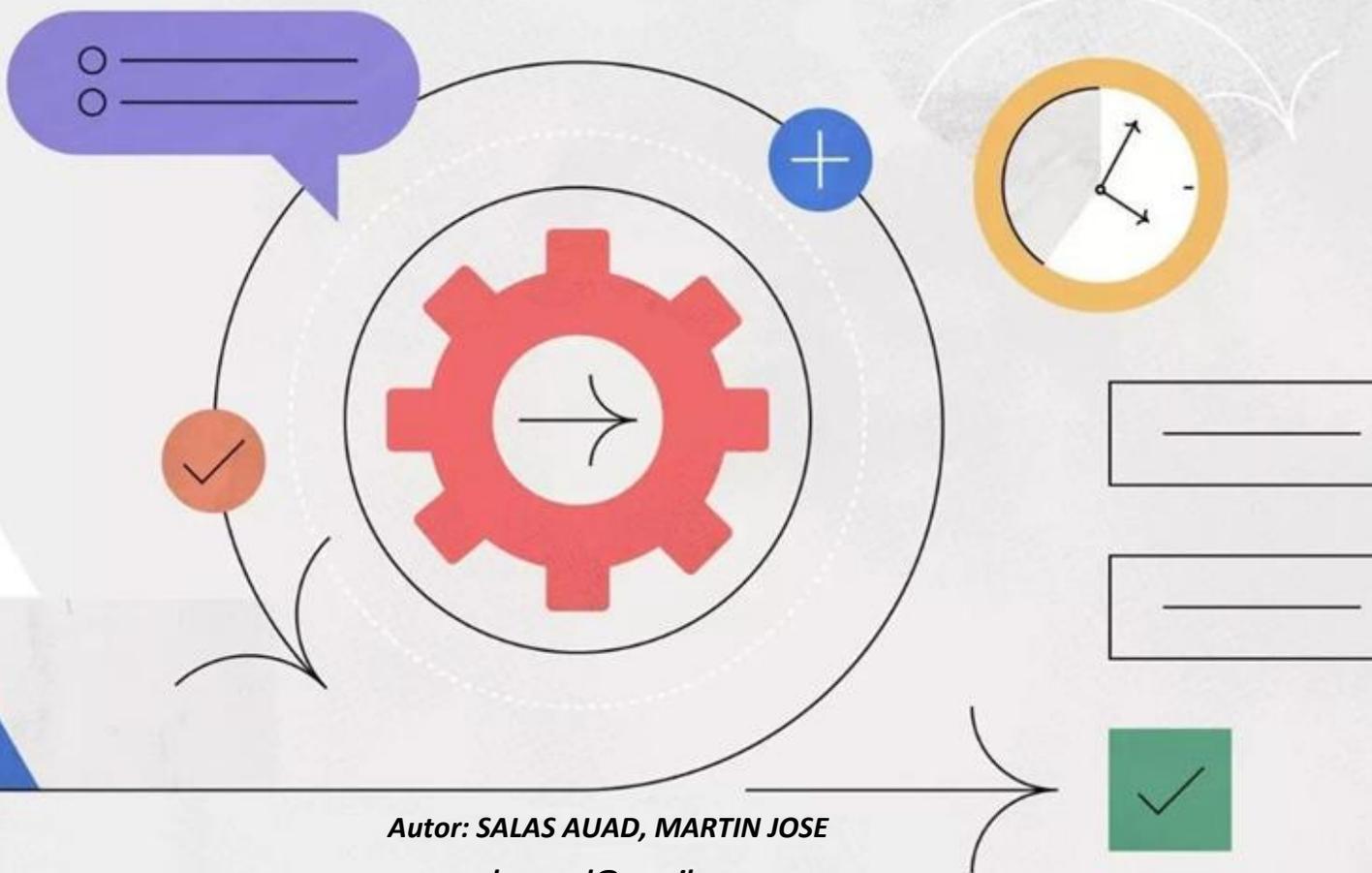




METODOLOGÍAS AGILES EN LA GESTIÓN DE PROYECTOS DE SOFTWARE EN UNA INSTITUCIÓN JUDICIAL



Autor: SALAS AUAD, MARTIN JOSE

salasauad@gmail.com

Tutor: Esteban Mulki

2024

Modalidad de la PP: Trabajo de aplicación de conceptos y técnicas de administración en situación laboral o ambiente real.



Contenido

Resumen	4
Introducción	6
Situación Problemática	6
Preguntas de Investigación.....	7
Objetivo General.....	7
Objetivos Específicos	8
Marco Metodológico	8
Enfoque	8
Diseño de Investigación	8
Técnicas de Recolección de Datos	8
Técnicas de Análisis.....	9
Marco Teórico.....	9
Introducción a las Metodologías Ágiles en el Desarrollo de Software	9
Scrum: Una Metodología Ágil para la Gestión de Proyectos.....	10
Kanban: Visualización y Optimización del Flujo de Trabajo	11
Scrumban: Un Enfoque Híbrido para la Gestión Ágil de Proyectos.....	11
Gestión Ágil de Producto: Priorización y Alineación con los Stakeholders	12
Profundización en la Mejora Continua (Scrum).....	13
Ventajas y Desafíos de la Implementación de Metodologías Ágiles en Instituciones Públicas	13
Integración de Otras Herramientas Ágiles.....	13
Aplicación	14
Primera Etapa: Análisis De La Situación Actual	14
Mapeo de Procesos.....	16
Segunda Etapa: Implementación De Metodologías Ágiles.....	17
Tercera Etapa: Evaluación Y Comparación	22
Recomendaciones.....	26
Conclusiones.....	27



Referencias	28
Apéndice	28
Guion de Entrevista Semi-estructurada:	28
Agenda de Capacitación.....	30
Agenda de Reunión con Stakeholders	31
Entrevista a miembros del equipo de desarrollo post-implementacion:.....	33
Anexo	34



Resumen

Este trabajo trata sobre la implementación de **metodologías ágiles** en la gestión de proyectos de desarrollo de software dentro del **Institución Judicial de Tucumán**, específicamente en la **Oficina de Desarrollo Tecnológico**, que pertenece al **Departamento de Tecnología**. La Oficina de Desarrollo Tecnológico es responsable de la creación y soporte de sistemas de software, brindando asistencia al **Equipo Científico**, un departamento crucial dentro de la Institución.

La gestión de proyectos en el **Institución Judicial de Tucumán** enfrenta desafíos que afectan la eficiencia y calidad del desarrollo de software. La falta de una metodología estructurada genera descoordinación y confusión en la asignación de tareas. Además, la comunicación limitada con usuarios finales y stakeholders provoca malentendidos y retrabajo, ya que algunas funcionalidades no se ajustan a sus necesidades reales.

El seguimiento inadecuado de los procesos y la ausencia de criterios claros para priorizar desarrollos contribuyen a retrasos y dificultades para gestionar el progreso. La falta de un enfoque ágil que se adapte a este contexto impide una alineación efectiva entre los objetivos institucionales y las entregas.

La adopción de metodologías ágiles, como **Scrum** y **Kanban**, fomenta una mejor organización, colaboración y seguimiento continuo, mejorando la calidad de los productos y la satisfacción del cliente interno.

Se realizaron estudios sobre metodologías ágiles, puntualmente Scrum y Kanban, y cuales son los beneficios de su utilización en instituciones del entorno gubernamental, que se caracteriza por procesos **burocráticos** y **estructurados**.

En la fase de análisis, se realizó un **diagnóstico** exhaustivo de los procesos y dinámicas de trabajo dentro de la Oficina de Desarrollo Tecnológico, identificando áreas clave de mejora. Para llevar a cabo este análisis se utilizaron distintas técnicas, como encuestas, entrevistas y observación participante.

La fase de **implementación** se centró en la adopción de las metodologías, para planificar de manera eficiente las tareas de desarrollo, realizar estimaciones aplicando técnicas como planning poker, organizar mejor la capacidad de trabajo del equipo, e involucrar activamente a los usuarios y stakeholders. Para conseguir esto se capacito al equipo en las nuevas metodologías, se definieron roles y se estableció un nuevo flujo de trabajo implementando los ciclos de desarrollo denominados Sprint.

En la fase de final se **evalúa** el impacto de las metodologías y herramientas ágiles aplicadas. Para esto se tuvo en cuenta los datos recolectados en cada ciclo de desarrollo. Esto permitió medir la **capacidad** del equipo, su **productividad** y **evolución** en el tiempo.



El resultado fue satisfactorio, se logró un cambio en la organización interna de la oficina, un mayor involucramiento de las partes interesadas y un incremento medible de la productividad del equipo.

Palabras Clave: Metodologías ágiles, gestión de proyectos, Scrum, desarrollo de software, institución judicial.



Introducción

En un entorno donde la **tecnología** es cada vez más central en la gestión de instituciones públicas, la eficiencia en los **proyectos de desarrollo de software** se ha vuelto un reto fundamental para asegurar la competitividad y la capacidad de respuesta. Este desafío es particularmente evidente en las **instituciones judiciales**, donde los sistemas deben garantizar la **precisión, seguridad y confiabilidad** para sostener procesos legales complejos.

La **Institución Judicial de Tucumán**, a través de su **Oficina de Desarrollo Tecnológico**, enfrenta la presión constante de entregar soluciones de software efectivas y adaptables a los cambios rápidos en los requisitos normativos y judiciales. La necesidad de implementar sistemas que respondan a las demandas de los stakeholders y, al mismo tiempo, cumplir con los plazos establecidos ha llevado a la adopción de enfoques más flexibles como las **metodologías ágiles**.

Las metodologías ágiles, como **Scrum** y **Kanban**, se han convertido en herramientas clave para enfrentar estos desafíos, ya que promueven la **colaboración**, la **adaptabilidad** y la **entrega continua de valor**. Diversos estudios han demostrado que la implementación de estas metodologías puede reducir los tiempos de entrega en proyectos de software en hasta un 25% y mejorar la satisfacción de los usuarios en más de un 30% (MIT Center for Digital Business, 2019). En el contexto judicial, donde los requisitos pueden cambiar rápidamente, estas metodologías permiten una mayor transparencia y una comunicación más fluida entre los desarrolladores y los stakeholders.

Este trabajo presenta un plan estructurado en tres fases: **análisis, implementación y evaluación**. En la primera fase, se llevará a cabo un diagnóstico detallado de los procesos actuales dentro de la Oficina de Desarrollo Tecnológico para identificar los principales cuellos de botella y áreas de mejora. Posteriormente, se implementarán metodologías ágiles, enfocándose en optimizar la **organización, colaboración y priorización** de tareas dentro del equipo de desarrollo. Finalmente, en la fase de evaluación, se analizarán los resultados de la implementación, comparando el estado previo y posterior al cambio, con el objetivo de establecer una base sólida para futuras mejoras continuas en los proyectos de software de la institución.

Situación Problemática

La gestión de proyectos de desarrollo de software en el **Institución Judicial de la provincia de Tucumán**, especialmente en el **Equipo Científico**, enfrenta varios desafíos significativos que comprometen la eficiencia y la calidad de los entregables. Uno de los problemas más destacados es la **falta de una metodología estructurada** para la organización de las tareas, lo que genera confusión y descoordinación entre los miembros del equipo.

La **colaboración limitada** entre los desarrolladores y otros actores clave, como los usuarios finales y los stakeholders, ha dado lugar a una comunicación deficiente y a



malentendidos sobre las expectativas y requisitos del software. Por ejemplo, en ocasiones, las funcionalidades desarrolladas no se alinean con las necesidades reales de los usuarios, lo que genera retrabajo.

Además, un **seguimiento inadecuado de los procesos** contribuye a retrasos en la entrega de proyectos. Sin un marco de trabajo claro, es difícil evaluar el progreso y gestionar las prioridades de manera efectiva. Estos problemas se ven agravados por la falta de un enfoque claro sobre cómo **priorizar** y decidir qué desarrollos se deben llevar a cabo. La ausencia de una metodología ágil impide una alineación efectiva entre las necesidades de la institución y las entregas de producto.

La implementación de metodologías ágiles, como **Scrum** y **Kanban**, podría abordar estos desafíos al promover una mejor organización del trabajo, fomentar la colaboración entre los equipos y facilitar un seguimiento constante de los procesos. Esto no solo mejoraría la calidad de los entregables, sino que también aumentaría la satisfacción del cliente interno, asegurando que las soluciones desarrolladas cumplan con los estándares y requisitos establecidos.

Preguntas de Investigación

1. ¿Cómo pueden las metodologías ágiles, como Scrum y Kanban, mejorar la colaboración y eficiencia en el desarrollo de software dentro de la **Oficina de Desarrollo Tecnológico**, que tiene como principal "cliente" al **Equipo Científico** de la **Institución Judicial de Tucumán**?
2. ¿Cuáles son los principales desafíos que enfrenta la **Oficina de Desarrollo Tecnológico** al implementar metodologías ágiles en un entorno judicial y multidisciplinario, considerando la dinámica con el Equipo Científico?
3. ¿Qué prácticas específicas de las metodologías ágiles resultan más efectivas para mejorar la priorización de tareas y la alineación con las necesidades del Equipo Científico en el contexto del desarrollo de software gubernamental?
4. ¿De qué manera puede la implementación de un marco de gestión ágil de producto optimizar la toma de decisiones y la priorización en los proyectos de desarrollo de software en la **Oficina de Desarrollo Tecnológico**?

Objetivo General

Proponer la adopción de metodologías ágiles en la gestión de proyectos de desarrollo de software dentro de la **Oficina de Desarrollo Tecnológico** perteneciente a una **Institución Judicial de la provincia de Tucumán**, con el objetivo de mejorar la organización, la colaboración y la eficiencia del trabajo del equipo, alineando las decisiones de priorización y desarrollo con las necesidades del **Equipo Científico**.



Objetivos Específicos

1. Identificar las metodologías ágiles más adecuadas para su aplicación en equipos de desarrollo de software dentro de la **Oficina de Desarrollo Tecnológico**, considerando las necesidades del **Equipo Científico**.
2. Proponer un plan de implementación de metodologías ágiles que sea adaptable a diferentes proyectos de programación y desarrollo dentro de la **Oficina de Desarrollo Tecnológico**.
3. Evaluar el impacto previsto de la adopción de metodologías ágiles en la eficiencia y calidad del trabajo en equipo de la **Oficina de Desarrollo Tecnológico**.
4. Definir un marco de gestión ágil de producto que facilite la toma de decisiones y la priorización de las tareas de desarrollo en la **Oficina de Desarrollo Tecnológico**, asegurando que las entregas de producto se alineen con las necesidades de los usuarios.

Marco Metodológico

Enfoque

Se emplea un enfoque mixto, combinando métodos cualitativos y cuantitativos para obtener una visión integral del impacto de la implementación de metodologías ágiles en la gestión de proyectos de desarrollo de software dentro de una institución judicial. Predomina un enfoque cualitativo, centrándose en la comprensión profunda de los procesos y dinámicas de trabajo, mientras que el enfoque cuantitativo se empleó para medir la eficiencia y los resultados obtenidos tras la implementación.

Diseño de Investigación

El diseño de la investigación sigue un ciclo iterativo de diagnóstico, planificación, acción y evaluación, propio de la investigación-acción. Este ciclo permite no solo intervenir directamente en el entorno de estudio para implementar y evaluar las metodologías ágiles, sino también medir el impacto de dicha implementación a través de indicadores cuantitativos.

Técnicas de Recolección de Datos

- **Cualitativas:**
 - **Entrevistas semi-estructuradas:** Se realizó con miembros del equipo de desarrollo, líderes de proyecto y otros actores clave, para captar percepciones, experiencias y desafíos relacionados con la gestión de proyectos antes y después de la implementación de metodologías ágiles.
 - **Observación participante:** Se observó las dinámicas de trabajo durante la implementación de las metodologías ágiles, identificando cambios, mejoras y áreas que requieren ajustes.



- **Revisión documental:** Se analizó la documentación existente relacionada con los procesos y resultados de proyectos anteriores, así como la documentación generada durante la implementación.
- **Cuantitativas:**
 - **Medición de tiempos y eficiencia:** Se utilizó indicadores de eficiencia, como el cumplimiento de plazos, la reducción de tiempos de entrega, y la cantidad de retrabajos o errores, para medir el impacto cuantitativo de la implementación de metodologías ágiles.
 - **Análisis de productividad:** A partir de los tiempos estimados para la entrega de nuevos módulos o proyectos, se comparan los resultados reales con los estimados para evaluar mejoras en la eficiencia tras la implementación.

Técnicas de Análisis

- **Análisis de contenido:** Para interpretar las entrevistas y observaciones, identificando patrones, temas recurrentes, y áreas de mejora en la dinámica de trabajo.
- **Análisis comparativo:** Para evaluar los cambios en la eficiencia y colaboración antes y después de la implementación, comparando datos cualitativos y cuantitativos.

Estadísticas descriptivas: Para cuantificar los resultados obtenidos y ofrecer una evaluación objetiva del impacto de las metodologías ágiles en la eficiencia del equipo.

Marco Teórico

Introducción a las Metodologías Ágiles en el Desarrollo de Software

Las **metodologías ágiles** surgieron a finales de la década de 1990 como una respuesta a los enfoques tradicionales de gestión de proyectos de software, que a menudo se caracterizaban por procesos rígidos, falta de flexibilidad ante cambios de requerimientos y largos tiempos de desarrollo antes de ver resultados tangibles. En 2001, un grupo de 17 expertos en desarrollo de software formuló el **Manifiesto Ágil**, que enfatizaba cuatro valores fundamentales: individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensa, colaboración con el cliente sobre la negociación contractual, y respuesta ante el cambio sobre seguir un plan.

A diferencia de las metodologías tradicionales, como el enfoque en cascada (Waterfall), las metodologías ágiles se basan en la entrega continua y el trabajo en pequeños incrementos, lo que permite al equipo de desarrollo adaptarse rápidamente a los cambios y reducir el riesgo. Las metodologías ágiles no son una técnica única, sino un conjunto de enfoques como **Scrum**, **Kanban** y **Extreme Programming (XP)** que comparten principios similares, pero con diferencias en su aplicación.



En entornos judiciales, donde las normativas pueden cambiar constantemente y las demandas de los stakeholders varían, las metodologías ágiles permiten responder rápidamente a estas fluctuaciones, mientras se asegura la calidad del desarrollo de software. **Schwaber y Sutherland (2014)** afirman que el uso de metodologías ágiles ha sido crucial en sectores regulados como el judicial, donde la adaptabilidad y la entrega rápida son esenciales para cumplir con las normativas en constante cambio. Un estudio realizado por el **MIT Center for Digital Business (2019)** reveló que la implementación de metodologías ágiles en proyectos gubernamentales ha permitido una reducción de hasta el 25% en los tiempos de entrega y un aumento del 30% en la satisfacción de los usuarios finales. Esta evidencia respalda la efectividad de las metodologías ágiles en mejorar la eficiencia y la calidad en entornos públicos.

Scrum: Una Metodología Ágil para la Gestión de Proyectos

Scrum, una de las metodologías ágiles más utilizadas, es particularmente adecuada para gestionar proyectos complejos donde se espera incertidumbre en los requisitos o el entorno del proyecto. El término fue acuñado por **Ken Schwaber** y **Jeff Sutherland** en la década de 1990, inspirado en el juego de rugby, donde los jugadores deben actuar en equipo de forma coordinada para avanzar hacia el objetivo. Scrum divide los proyectos en ciclos de trabajo cortos llamados **sprints** (normalmente entre 2 y 4 semanas), al final de los cuales se espera tener un incremento funcional del producto que puede revisarse y entregarse.

- **Roles en Scrum:**

- **Product Owner:** Es el encargado de maximizar el valor del producto. Define las prioridades en el **Product Backlog** y se asegura de que el equipo trabaje en lo más valioso para la organización.
- **Scrum Master:** Un facilitador que garantiza que el equipo de desarrollo siga las reglas de Scrum y supere los obstáculos que puedan surgir durante el sprint.
- **Equipo de Desarrollo:** Un grupo de profesionales multidisciplinarios que trabaja de manera autónoma para completar las tareas priorizadas durante el sprint.

Product Backlog: El **Product Backlog** es una lista priorizada de todas las tareas, funcionalidades y requisitos que se deben implementar en un proyecto. Este backlog es gestionado por el **Product Owner**, quien se asegura de que esté alineado con los objetivos de la organización y que las tareas estén organizadas de acuerdo a su prioridad. Cada elemento del backlog debe ser claro y conciso, facilitando al equipo de desarrollo la comprensión de lo que se debe realizar y el valor que cada tarea aporta. Es importante que el backlog sea un documento vivo, actualizado de manera continua a medida que se obtiene nueva información y feedback, permitiendo al equipo adaptarse a los cambios en los requisitos del proyecto. Según **Schwaber y Sutherland (2014)**, el Product Backlog es fundamental para el éxito de Scrum, ya que proporciona una base clara sobre la cual el equipo puede trabajar y tomar decisiones informadas a lo largo del desarrollo del producto.



Schwaber y Sutherland (2014) explican que **Scrum** no solo fomenta la entrega rápida de valor, sino que también ayuda a los equipos a adaptarse rápidamente a los cambios, ya que después de cada sprint se realiza una retrospectiva donde se identifican los puntos de mejora. Esta práctica permite mejorar constantemente tanto los procesos como el producto final. En una institución judicial, esta retroalimentación constante es crucial para asegurar que los desarrollos cumplan con las regulaciones cambiantes y las necesidades de los usuarios.

Kanban: Visualización y Optimización del Flujo de Trabajo

Mientras que Scrum trabaja en ciclos definidos, **Kanban** ofrece un enfoque más flexible. **Kanban** proviene de la industria manufacturera, específicamente del sistema de producción de **Toyota**, donde se utilizaba para controlar el inventario de producción. En el contexto del desarrollo de software, Kanban se utiliza para visualizar el flujo de trabajo y mejorar continuamente la eficiencia del equipo. Las tareas se representan como tarjetas en un tablero visual dividido en columnas como "Por hacer", "En progreso" y "Hecho". Este enfoque permite a los equipos ver claramente dónde se encuentran los cuellos de botella en el proceso y priorizar las tareas con base en la capacidad disponible.

Un estudio de **Pichler (2016)** resalta que Kanban es especialmente útil en organizaciones donde las tareas no siempre llegan de manera regular o donde los cambios de prioridad son comunes. En las instituciones judiciales, donde los desarrolladores de software deben reaccionar a nuevas leyes, actualizaciones de políticas o cambios en las prioridades gubernamentales, Kanban proporciona una herramienta visual que permite ajustar el flujo de trabajo de manera eficiente.

Scrumban: Un Enfoque Híbrido para la Gestión Ágil de Proyectos

Scrumban combina las mejores prácticas de Scrum y Kanban, proporcionando un enfoque híbrido para la gestión ágil de proyectos. Originalmente diseñado como un puente para equipos que deseaban migrar de Scrum a Kanban, Scrumban se ha convertido en una metodología con identidad propia, especialmente útil en contextos donde se necesita una estructura inicial que permita flexibilidad y optimización continua.

En Scrumban, los sprints de Scrum pueden ser opcionales o adaptables, priorizando la gestión visual del flujo de trabajo a través de un tablero Kanban. Este enfoque es particularmente efectivo en entornos como las instituciones judiciales, donde las demandas de los stakeholders pueden ser impredecibles y los cambios de prioridad son frecuentes.

Características principales de Scrumban:

1. **Gestión visual:** Utiliza tableros Kanban para visualizar el flujo de trabajo, identificar cuellos de botella y optimizar procesos.
2. **Mejora continua:** Se enfoca en la retrospectiva y la mejora constante del proceso, heredado de Scrum.
3. **Planificación por demanda:** En lugar de ciclos rígidos, las tareas se planifican a medida que se identifican necesidades o requisitos, proporcionando mayor flexibilidad.



4. **WIP (Work in Progress):** Limita el trabajo en progreso para evitar la sobrecarga del equipo y mejorar la calidad de las entregas.

Según Kniberg (2013), Scrumban es ideal para proyectos en entornos dinámicos, como las instituciones públicas, donde los equipos enfrentan flujos de trabajo variables y prioridades en constante cambio. En el Ministerio Público Fiscal, por ejemplo, podría implementarse para combinar la estructura de Scrum con la adaptabilidad visual de Kanban, permitiendo a los equipos responder rápidamente a cambios regulatorios y necesidades emergentes.

Un estudio realizado por el Lean Agile Institute (2020) mostró que la adopción de Scrumban en equipos de desarrollo permitió reducir los tiempos de entrega en un 20% y mejorar la satisfacción de los stakeholders en un 25%. Esto lo convierte en una opción viable para equipos que desean aprovechar los beneficios de ambos enfoques ágiles.

Gestión Ágil de Producto: Priorización y Alineación con los Stakeholders

La **gestión ágil de producto** es un componente clave que complementa las metodologías ágiles al asegurar que las funcionalidades desarrolladas estén alineadas con las necesidades y expectativas de los stakeholders. Este enfoque permite priorizar el desarrollo con base en el valor que cada funcionalidad aporta al usuario final, lo que es especialmente importante en proyectos judiciales, donde las necesidades de los diferentes actores (abogados, jueces, personal administrativo) pueden variar.

Patton (2014), en su libro *User Story Mapping*, argumenta que la gestión ágil de producto permite estructurar los desarrollos de manera que maximicen el valor entregado. Un **Product Backlog** bien gestionado es fundamental para asegurar que las decisiones técnicas y estratégicas estén alineadas con los objetivos de la organización. Esto se logra mediante la creación de **User Stories**, que permiten descomponer grandes objetivos en tareas más pequeñas, priorizando aquellas que aportan más valor.

Las **historias de usuario** son una herramienta clave en la gestión ágil de producto para definir y priorizar el trabajo en términos de valor para el usuario final. Estas se escriben desde la perspectiva del usuario y siguen un formato estructurado:

"Como [tipo de usuario], quiero [acción/funcionalidad] para [beneficio o propósito]."

Este formato ayuda a los equipos a mantener el enfoque en el usuario y sus necesidades reales, promoviendo la colaboración entre el equipo de desarrollo y los stakeholders. Las historias de usuario también permiten identificar claramente los entregables de cada sprint y medir su impacto en el proyecto.

En la gestión ágil de producto, las historias de usuario se priorizan en el **Product Backlog** en función de varios criterios:



- El valor que aportan al usuario o al negocio.
- La alineación con los objetivos estratégicos del proyecto.
- La complejidad técnica y el esfuerzo estimado para su implementación.

La priorización se realiza de manera colaborativa, liderada por el **Product Owner** con el apoyo de los stakeholders, asegurando que el desarrollo avance en la dirección correcta.

Esta práctica no solo mejora la planificación, sino que también permite una adaptación rápida a cambios en las prioridades, un aspecto crítico en instituciones judiciales, donde las necesidades pueden cambiar debido a nuevos marcos legales o regulaciones.

Profundización en la Mejora Continua (Scrum)

Una de las características que distingue a Scrum es su enfoque en la mejora continua. Al final de cada sprint, se lleva a cabo una **Sprint Retrospective**, donde el equipo de desarrollo reflexiona sobre el desempeño del sprint y analiza lo que se puede mejorar en el próximo ciclo. Esto fomenta un ciclo constante de optimización y ajuste, lo que resulta en un proceso de desarrollo más eficiente.

En el contexto judicial, este proceso es crucial, ya que permite realizar ajustes rápidos en respuesta a cambios regulatorios o necesidades emergentes. **Sutherland (2014)** destaca que las retrospectivas no solo ayudan a mejorar la dinámica del equipo, sino que también ofrecen la oportunidad de recibir retroalimentación crítica de los stakeholders, asegurando que los desarrollos se ajusten a sus expectativas y requisitos legales.

Ventajas y Desafíos de la Implementación de Metodologías Ágiles en Instituciones Públicas

La implementación de metodologías ágiles en instituciones públicas presenta una serie de desafíos específicos. Las barreras burocráticas, la resistencia al cambio y las limitaciones presupuestarias son obstáculos comunes que deben superarse. Sin embargo, las instituciones públicas también se benefician significativamente de la agilidad, ya que estas metodologías permiten una mayor adaptabilidad a los cambios normativos y políticos.

Un estudio de **MIT Center for Digital Business (2019)** sobre proyectos gubernamentales en los Estados Unidos mostró que la implementación de Scrum y Kanban en proyectos de software permitió reducir los tiempos de entrega en un 25%, mejorando al mismo tiempo la satisfacción de los usuarios finales. Estos resultados reflejan cómo las metodologías ágiles pueden ayudar a superar las ineficiencias asociadas a las prácticas tradicionales en entornos públicos.

Integración de Otras Herramientas Ágiles

Las herramientas tecnológicas también juegan un papel importante en la implementación de metodologías ágiles. **Jira**, **Trello** y **Confluence** son algunas de las herramientas más comunes que permiten a los equipos gestionar sus tareas y proyectos de manera eficiente. En un contexto judicial, donde la precisión es fundamental, estas



herramientas proporcionan una manera clara de hacer un seguimiento del trabajo en curso, gestionar las prioridades y garantizar que los plazos se cumplan.

Jira, por ejemplo, permite a los equipos gestionar el **Product Backlog**, asignar tareas y hacer un seguimiento detallado del progreso del trabajo. Esta herramienta es especialmente útil en la gestión ágil de producto, ya que proporciona un entorno transparente donde los stakeholders pueden ver el estado del desarrollo en tiempo real.

Estudios y Ejemplos Relevantes

El uso de metodologías ágiles en instituciones públicas ha mostrado resultados positivos en diversos países. En Suecia, un estudio realizado en 2018 mostró cómo la adopción de **Kanban** en el sistema judicial permitió reducir los tiempos de respuesta en un 40%, mejorando la eficiencia entre diferentes departamentos. En Nueva Zelanda, la implementación de **Scrum** en proyectos gubernamentales resultó en una mejor alineación entre los equipos de desarrollo y los stakeholders, aumentando la entrega de productos funcionales en un 30%.

En Noruega, un equipo de desarrollo de software en el sector público adoptó Scrumban para gestionar proyectos de tecnología educativa. Según un estudio de Agile Norway (2021), esta implementación permitió reducir las interrupciones no planificadas en un 30%, mejorando al mismo tiempo la transparencia del flujo de trabajo entre los departamentos involucrados.

Aplicación

Primera Etapa: Análisis De La Situación Actual

En la primera etapa se priorizó la recolección y revisión de documentación perteneciente al Departamento de Tecnología de la Institución Judicial, se realizó una encuesta a través de un formulario de Google a todos los miembros del Departamento de Tecnología, se entrevistó al jefe del Departamento y se recolectó información de procesos y metodologías a través de observación participante. Se enfocaron los esfuerzos principalmente en la **Oficina de Desarrollo Tecnológico (DT)**.

Se realizó un relevamiento de la estructura organización del Departamento de Tecnología para entender la posición de la oficina de Desarrollo Tecnológico dentro del Departamento, como así también las relaciones y dependencias que existen entre las mismas.

Gráfico Nº 1: Organigrama Departamento de Tecnología



Fuente: elaboración propia

El Departamento de Tecnología cuenta con tres (3) equipos de desarrollos, cada uno de los equipos opera de manera independiente. Esta es una de las primeras alertas detectadas en este estudio, ya que anuncia posibles dificultades a la hora de interactuar entre los equipos, que, si bien todas responden a un mismo jefe de departamento, al ser independientes, cada uno de los equipos cuenta con su propia estructura organizativa y establece sus propias metodologías de trabajo.

Cada equipo trabaja con usuarios y en sistemas diferentes, y todos los sistemas son desarrollos propios de los equipos pertenecientes a la Institución Judicial.

El equipo de Desarrollo 1 se enfoca en el manejo de legajos y todo lo orientado a la parte legal de la Institución Judicial, también tiene a su cargo el sistema que utiliza el área de Recursos Humanos y el sistema de manejo de inventarios de la institución. Está conformado por 14 personas, es el equipo más grande con el que cuenta el departamento, la mayor parte del mismo son desarrolladores (10), un coordinador del equipo, un jefe interno y 2 personas dedicadas exclusivamente a brindar soporte técnico a los usuarios de los sistemas.

El equipo de Desarrollo 2, en el cual nos enfocaremos en este trabajo, se denomina **Oficina de Desarrollo Tecnológico (DT)**. La misma se encarga de asistir al Equipo Científico de la Institución Judicial. El Equipo Científico es uno de los departamentos más grandes e importantes dentro de la Institución y cuyas funcionalidades y procesos son críticos para el cumplimiento de los objetivos de la organización. **DT** se encarga de desarrollar y mantener dos sistemas fundamentales, **sistema1** utilizado para realizar investigaciones de antecedentes y otro, al cual nombraremos **sistema2** cuya función principal es la de gestión de muestras científicas, pruebas y pericias relevantes para llevar a cabo las investigaciones judiciales. **DT** está compuesto por 5 miembros, todos desarrolladores, uno de ellos, el que cuenta con mayor antigüedad y experiencia es el coordinador y encargado del equipo.

En este trabajo se utilizará como prueba piloto de implementación el desarrollo de un módulo nuevo para el **sistema2**, que promete agilizar tareas de recolección y seguimiento de pruebas, al mismo lo nombraremos como **Proyecto Piloto**.



El equipo de desarrollo 3, se encarga de los desarrollos de uso externo a la Institución Judicial y los sistemas de seguridad internos. Cuenta con 3 miembros, todos desarrolladores, uno de ellos cumple la función de coordinador.

El Área de Sistemas cuenta con 8 miembros y cumple la función de brindar soporte técnico, manejo de servidores y realizar mantenimiento a los equipos de toda la institución Judicial, incluidos los equipos de desarrollo.

El Área de Monitoreo cuenta con 2 miembros, está vinculada con el equipo de desarrollo 3. Y cumple la función de monitoreo de cámaras de seguridad de la Institución.

Mapeo de Procesos

A través de la observación participante, fue posible relevar los procesos que se aplican dentro de la Oficina de Desarrollo Tecnológico (DT) y elaborar un diagrama de flujo que detalla el flujo de trabajo actual. Este representa el recorrido desde la recepción de un nuevo requerimiento de desarrollo hasta su finalización.

Gráfico N° 2: Flujo de trabajo inicial



Fuente: elaboración propia

El flujo de trabajo observado sigue una dinámica lineal. Esto significa que, una vez recibido el requerimiento, generalmente como una directiva proveniente de los altos mandos y con poca especificación técnica o funcional, se realiza un análisis de viabilidad seguido de una planificación general. Actualmente, esta planificación se lleva a cabo utilizando la metodología en cascada (Waterfall).

La metodología en cascada es un enfoque tradicional para el desarrollo de software que organiza el trabajo en etapas secuenciales y rígidas. Estas etapas deben completarse antes de avanzar a la siguiente. Este modelo es altamente estructurado y requiere definir de manera precisa el alcance, los tiempos y las metas desde el inicio del proyecto, su principal desventaja es la rigidez: los cambios en los requerimientos durante el desarrollo son difíciles de gestionar. En el caso de DT, los tiempos y metas están directamente vinculados a las fechas límites establecidas en los requerimientos, lo que puede limitar la flexibilidad para adaptarse a necesidades emergentes o imprevistos.



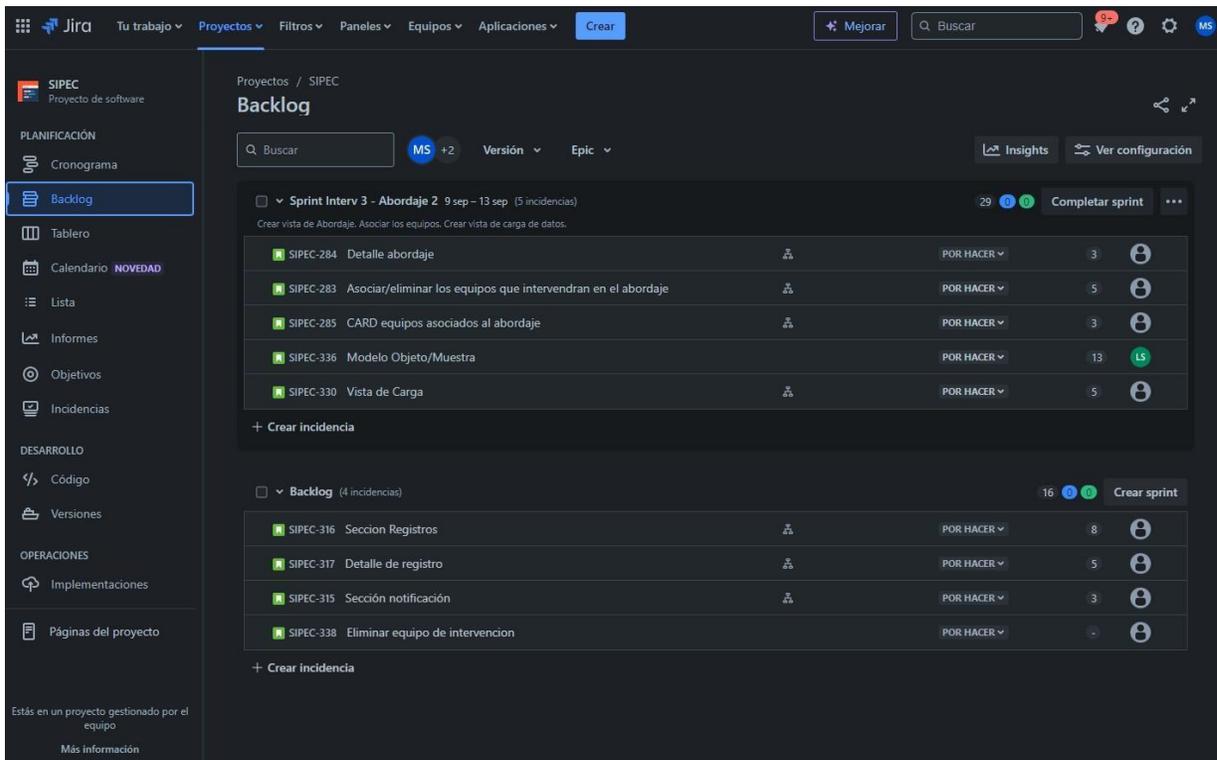
Para la gestión y planificación del desarrollo el equipo de DT utiliza la herramienta JIRA, si bien la misma cuenta con muchas funcionalidades que permiten una organización eficiente de las tareas a cumplir dentro de un equipo, en DT solo se la utilizaba para realizar un registro de las tareas, a modo de recordatorio. Sin un orden o priorización de actividades, ni una estructura clara de cómo se registran dichas tareas. Esto genera ineficiencias, registros dobles, incertidumbre en los tiempos e incumplimiento de las fechas establecidas en la planificación inicial.

Segunda Etapa: Implementación De Metodologías Ágiles

La segunda etapa del proyecto se enfocó en la capacitación de los miembros del equipo de Desarrollo Tecnológico (DT). Durante esta instancia, se abordaron los principios fundamentales de las metodologías ágiles, con énfasis en Scrum y Kanban, destacando cómo ambas metodologías pueden complementarse para optimizar los procesos.

La capacitación se desarrolló con base en una agenda, que incluyó el uso de diapositivas ilustrativas para facilitar la comprensión de los conceptos. Además, se realizó un recorrido práctico por la herramienta JIRA, haciendo especial énfasis en la gestión del **Backlog** y en el uso del tablero **Kanban** para visualizar y organizar las tareas de manera eficiente.

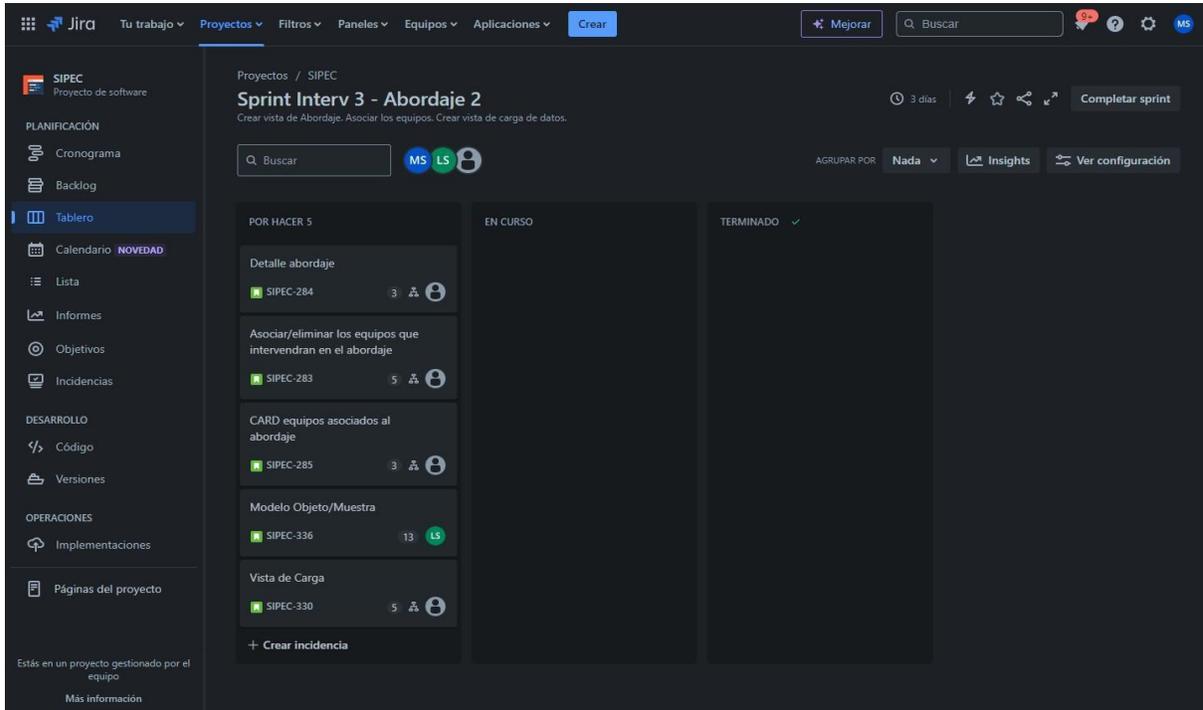
Imagen N° 1: Backlog del proyecto piloto



Fuente: elaboración propia



Imagen N° 2: Tablero Kanban

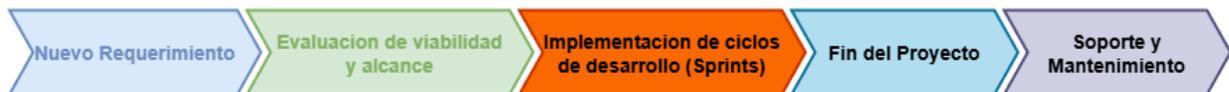


Fuente: elaboración propia

Como cierre de la capacitación, se llevó a cabo una actividad práctica en la que el equipo trabajó en la elaboración del backlog inicial del proyecto piloto. Esto incluyó la creación de historias de usuario y la estimación de esfuerzo para cada una de ellas. Asimismo, se definió que la duración de los Sprints sería de una (1) semana, y se llevó a cabo la reunión de planificación correspondiente al **Sprint 0**, marcando el inicio formal del trabajo bajo las metodologías ágiles.

Se planteó un nuevo flujo tentativo de trabajo:

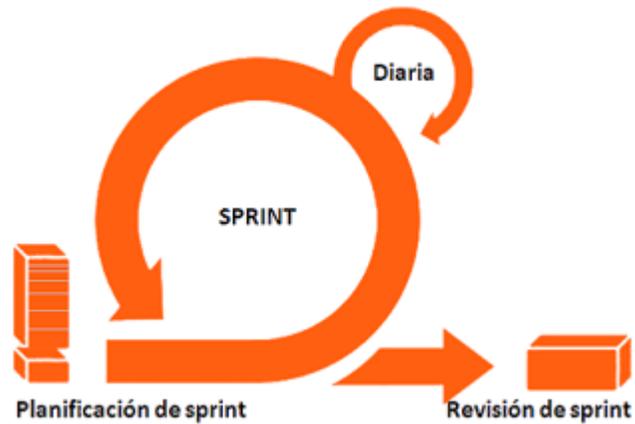
Gráfico N° 3: Flujo de trabajo propuesto



Fuente: elaboración propia



Gráfico Nº 4: Ilustración de un Sprint

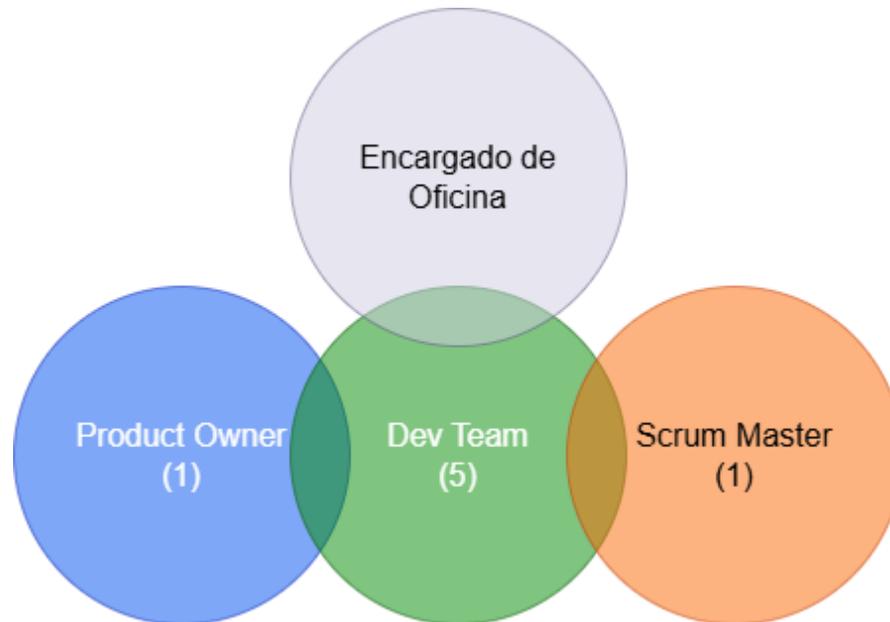


Fuente: <https://scrum.menzinsky.com/2017/01/como-imprimen-los-sprints-flujo-en-la.html>

Como parte de la implementación, se estableció una nueva estructura dentro de DT acorde con la metodología Scrum, asignando roles específicos a los miembros del equipo. Uno de los integrantes asumió el rol de **Product Owner**, siendo responsable de recopilar y definir los requisitos para el desarrollo del proyecto piloto. El autor de este trabajo desempeñó la función de **Scrum Master**, facilitando la comunicación y supervisando los procesos del equipo. El **Equipo de Desarrollo** estuvo compuesto por cinco miembros, incluyendo al responsable de la oficina, quien redujo sus responsabilidades en la gestión de proyectos para enfocarse en el desarrollo del piloto.



Gráfico Nº 5: Ilustración de funciones dentro de DT



Fuente: Elaboración propia

Durante el desarrollo del segundo sprint (**Sprint 1**) del proyecto piloto, el miembro del equipo que desempeñaba el rol de **Product Owner** renunció a su puesto, dejando la posición vacante. Ante la imposibilidad de asignar el rol a otro integrante del equipo, el autor del trabajo, que ya cumplía la función de **Scrum Master**, tuvo que asumir temporalmente algunas de las responsabilidades del **Product Owner**, como la gestión del backlog, la interacción con los stakeholders y la priorización de objetivos. Esta situación generó un conflicto de intereses, ya que ambas funciones tienen enfoques diferentes: el **Scrum Master** se encarga de facilitar el proceso y proteger la metodología, mientras que el **Product Owner** se centra en la visión del producto y en definir las prioridades del proyecto. A su vez, la disminución de los miembros del equipo generó una reducción en la capacidad de desarrollo del equipo.

En la implementación del tercer ciclo de desarrollo (**Sprint 2**), surgieron distintas situaciones que interrumpieron el flujo establecido de trabajo. Se recibieron solicitudes de los mandos altos de la institución para realizar modificaciones de sistemas que ya se encuentran en funcionamiento, lo que requiere la disposición de capacidad de desarrollo que inicialmente estaba asignada al ciclo de desarrollo del proyecto piloto, lo cual compromete el alcance pactado en la planificación de los Sprints y se vio reflejado en el rendimiento del ciclo.

El **sprint 3** se desarrolló con normalidad, no hubo incidencias. Se intentó pactar una reunión con los stakeholders, pero no fue posible llevarla a cabo. Al finalizar el mismo, se realizó la primera reunión de retrospectiva con los miembros del equipo, en la cual se



estableció dejar de aplicar las reuniones diarias, práctica fundamental en la metodología Scrum, pero que dado el contexto en el que se desarrolla la actividad perdía relevancia ya que todos los miembros del equipo trabajan en un mismo espacio físico y la comunicación fluye durante toda la jornada de trabajo, permitiendo saber en qué está enfocándose cada uno de los miembros en todo momento y resolviendo las dificultades en equipo apenas son detectadas.

El rendimiento del **Sprint 4** se vio afectado por la desvinculación repentina de uno de los miembros del equipo de desarrollo, el cual quedó conformado por 3 personas. Esto generó un ambiente de incertidumbre en el grupo y un impacto negativo en la motivación de los integrantes. La productividad disminuyó durante el ciclo y las tareas que no se llegaron a completar en el mismo se pasaron al ciclo de desarrollo siguiente.

Durante el **Sprint 5** se realizaron cambios en cuanto a la duración de los ciclos, se pasó de 1 semana a 2 semanas de duración. Esto permitió enfocar en mayor medida los esfuerzos en el cumplimiento de las tareas y amplió el margen de tiempo para la organización de reuniones con los Stakeholders, una de las barreras más predominantes durante la ejecución de la nueva dinámica de trabajo, debido a que la mayoría de los interesados cumplen funciones que imposibilitan la participación activa en el desarrollo del proyecto. Si bien la reunión se pactó para el final del Sprint 5, se pudo llevar a cabo a mediados del Sprint 6, en la misma participaron 14 personas, entre ellos se encontraban jefes de los equipos científicos, como también futuros usuarios del sistema. Se elaboró una agenda de reunión como herramienta de apoyo y se establecieron los objetivos deseados de la misma, el objetivo principal era conseguir un mayor involucramiento. La reunión tuvo una repercusión positiva, posibilitó la obtención de retroalimentación sobre lo ya desarrollado hasta el momento y sugerencias sobre nuevas funcionalidades que se podrían contemplar en ciclos futuros.

A partir del feedback obtenido en la reunión, fue posible ajustar la planificación para los siguientes Sprints, a su vez, se logró entablar canales de comunicación con algunos de los stakeholders quienes se comprometieron a participar de forma activa en futuras reuniones. Durante el **Sprint 6** y **7** surgieron nuevas dificultades, si bien se consiguió una mayor participación de algunos stakeholders y se logró realizar una nueva reunión, el desarrollo se vio retrasado por el incumplimiento de compromisos tomados por los stakeholders, quienes debían aportar una serie de documentación que brindan información sobre el marco legal que tiene que respetar los documentos generados por el nuevo sistema. Teniendo en cuenta estos retrasos, se destinaron los esfuerzos en finalizar otras funcionalidades del sistema y se iniciaron los trabajos de integración de los sistemas. En este punto, el nuevo sistema se encuentra finalizado en un 87% (realización del backlog), requiere completar los ajustes finales para la realización de pruebas de campo. Las pruebas de campo implican trabajar a la par de los equipos científicos en la recolección de pruebas, generando una carga paralela de información, evaluando el desempeño del sistema y la capacidad del mismo de reemplazar las metodologías de recolección previas.



Tercera Etapa: Evaluación Y Comparación

Se realizó una recopilación de los datos obtenidos de cada ciclo de desarrollo, de esta manera fue posible evaluar el progreso del desarrollo, la velocidad del equipo y las tasas de realización.

Para obtener estos datos la herramienta JIRA cuenta con una sección para generar informes que permiten evaluar todos estos factores, pero por cuestiones de configuración de la herramienta el apartado de informes no fue de utilidad. Para suplantar estos informes se optó por la utilización de una planilla de Excel en la cual se trasladaron todos los datos recolectados en cada Sprint.

En la misma se incluyen datos relevantes de cada Sprint, como la duración, cantidad de miembros participantes, incidencias comprometidas, puntos de historias, porcentaje de cumplimiento, rendimiento comparado con el Sprint 0 y el rendimiento comparado con el Sprint anterior.

Tabla N° 1: Tabla de seguimiento de Sprints

N° Sprint	Duración(semanas)	Miembros activ	Incidencias	Story Points	Completado	Completado %	Rendimiento(base S-0)	Rendimiento(base S-previo)
0	1	5	10	20	20	100%	100%	
1	1	4	6	28	28	100%	140%	140%
2	1	4	7	37	11	30%	55%	39%
3	1	4	8	33	33	100%	165%	300%
4	1	4	7	37	16	43%	80%	48%
5	2	3	12	69	53	77%	265%	331%
6	2	3	7	42	16	38%	80%	30%
7	2	3	11	49	18	37%	90%	113%
8	2	3	4	31	-	-	-	-

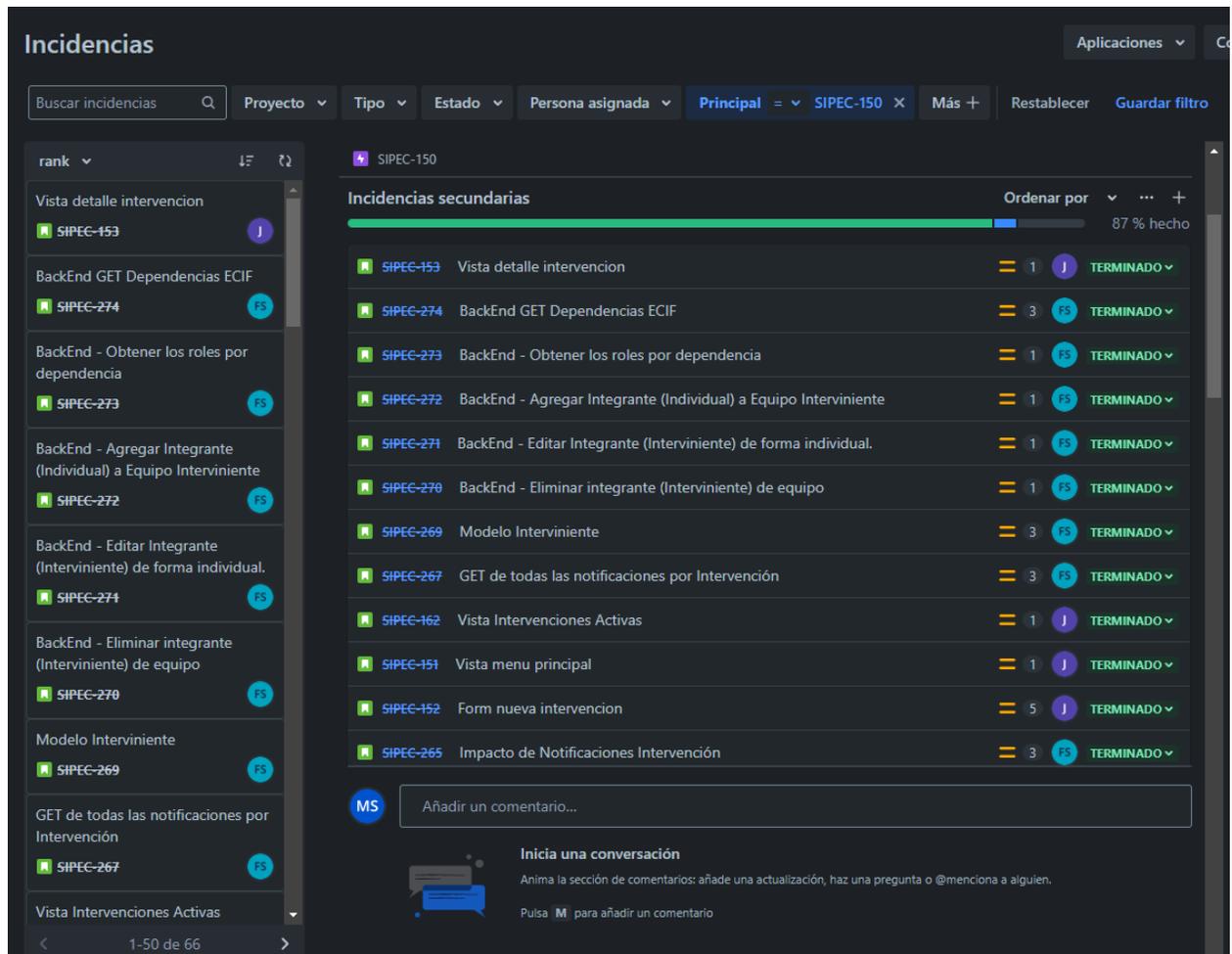
Fuente: Elaboración propia.

A partir de los datos recolectados es posible determinar una velocidad promedio del equipo por sprint de 24 puntos, para obtener este valor se sacó un promedio de los puntos de historia completados en cada sprint y se los dividió en la cantidad de ciclos.

El porcentaje de finalización del backlog se encuentra en un 87%, dato que se obtiene del tablero de Incidencias de la plataforma JIRA.



Imagen Nº 3: Captura de pantalla del progreso del Backlog en la etapa final del proyecto.



Fuente: Elaboración propia.

Todos los desarrollos previos al proyecto no aplicaban metodología de medición de avances, se cuenta únicamente con las fechas de inicio y finalización de incidencias cargadas en la herramienta JIRA, lo cual dificulta una comparación cuantitativa de los resultados.

Si fue posible determinar el incremento en la eficiencia entre ciclos de desarrollo dentro del mismo proyecto, y se puede destacar el gran impacto que genera en él productividad las incidencias externas.



Tabla Nº 2: Tabla registro de sucesos.

N° Sprint	OBSERVACIONES
0	Capacitación introductoria
1	Renuncia 1 miembro del equipo.
2	Solicitud adaptación otros sistemas / 1 miembro menos la mitad del sprint (licencia)
3	No hubo incidencias
4	Desvinculación de un miembro del equipo / Incertidumbre en el equipo
5	Reunión retrospectiva 8-10 / sprint pasa a 2 semanas
6	Interrupcion de desarrollo - Nuevas prioridades / Reunión de demostración con Stakeholders
7	Incumplimientos externos a DT / Reunión de retroalimentación con Stakeholders
8	El Sprint no se inicio - No es posible avanzar debido a la dependencia de factores externos a la oficina

Fuente: Elaboración propia.

Los ciclos 1, 3 y 5, los cuales no sufrieron interrupciones, mostraron un incremento significativo en la productividad con respecto al ciclo inicial.

Tabla Nº 2: Tabla comparativa de rendimientos entre los ciclos.

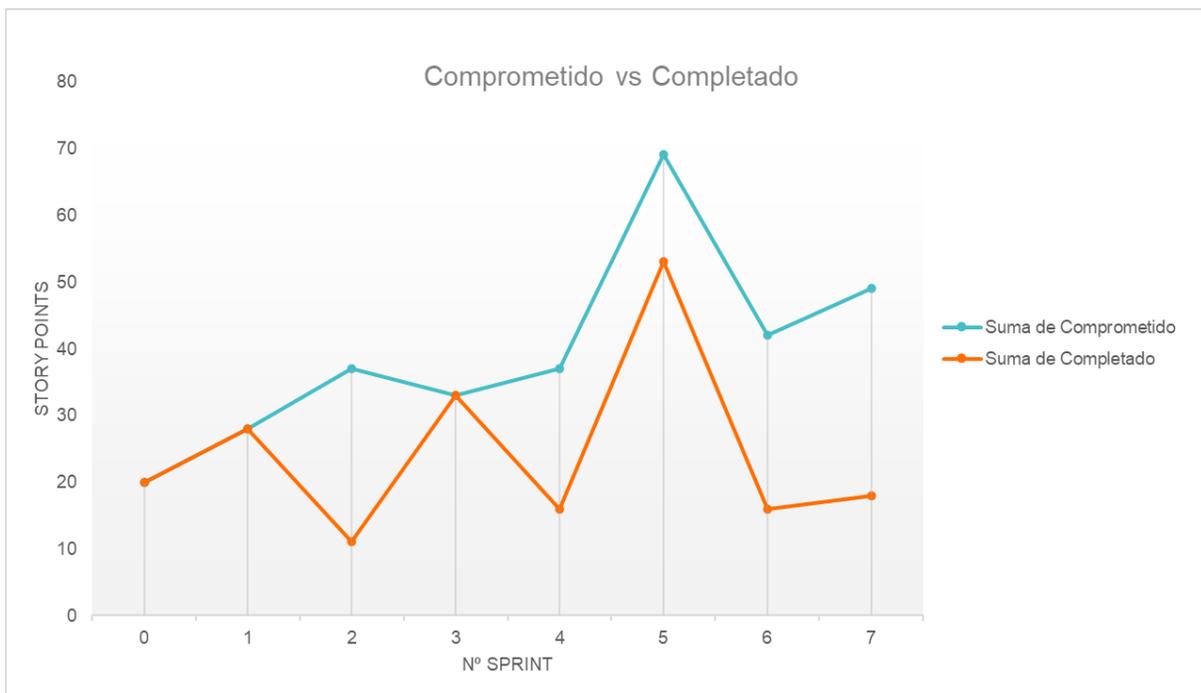
N° Sprint	Story Points	Completado	Completado %	Rendimiento (base S-0)
0	20	20	100%	100%
1	28	28	100%	140%
2	37	11	30%	55%
3	33	33	100%	165%
4	37	16	43%	80%



5	69	53	77%	265%
6	42	16	38%	80%
7	49	18	37%	90%

Fuente: Elaboración propia.

Gráfico Nº 6: Comparativa de Story points comprometidos y completados

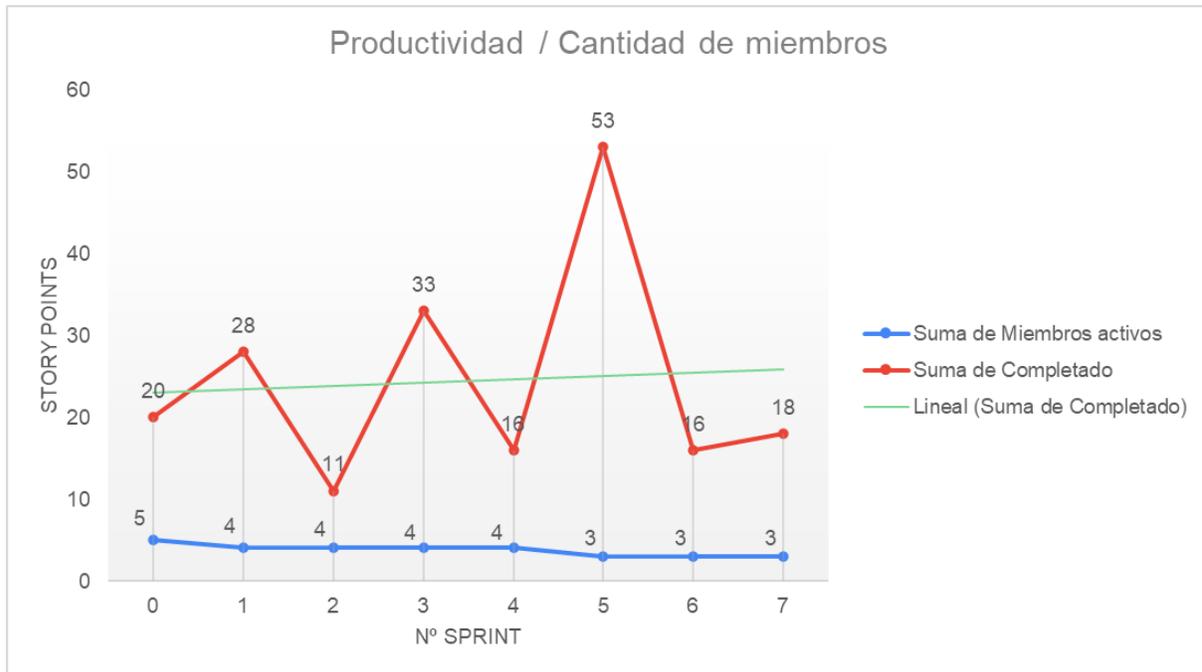


Fuente: elaboración propia.

En el gráfico anterior se pueden detectar que las líneas solo se solapan en 3 puntos, en los sprint 0, 1 y 3, esto implica que la totalidad de lo planificado dentro del ciclo pudo llevarse a cabo dentro del plazo establecido.



Gráfico N° 7: Evolución de la productividad / Miembros activos del equipo



Fuente: Elaboración propia.

En el gráfico 7, se puede verificar la relación entre la cantidad de los miembros del equipo y la productividad. En verde se marcó la línea de tendencia de la productividad, mostrando un crecimiento a pesar de disminuir la cantidad de miembros activos en el equipo.

Recomendaciones

Capacitación continua del equipo:

Se recomienda realizar capacitaciones más profundas para los miembros del equipo de desarrollo, enfocadas en el uso y los beneficios de herramientas ágiles como la planificación, retrospectivas y el tablero Kanban. Esto permitiría una mejor comprensión y aplicación de estas herramientas.

Formación para stakeholders:

Es importante ofrecer capacitaciones sobre metodologías ágiles a todos los stakeholders involucrados en futuros proyectos. Esto fomentará una mayor comprensión de su rol y la importancia de su participación en el proceso, mejorando la colaboración y el desarrollo de productos más ajustados a sus necesidades.

Mejora en la recopilación de requerimientos:

Se sugiere trabajar en la creación de canales más efectivos para la recopilación de requisitos desde el inicio de los proyectos, asegurando que estos estén bien definidos y



basados en información concreta, evitando suposiciones. Esto podría incluir sesiones de trabajo conjunto con los stakeholders y el uso de técnicas como entrevistas y talleres de co-creación.

Fortalecimiento de la comunicación externa:

Mejorar la comunicación con agentes externos es clave para incrementar la transparencia y facilitar la participación de los stakeholders. Esto podría lograrse mediante reuniones periódicas, reportes de avances y la integración de plataformas de comunicación colaborativa.

Mantener el uso de herramientas clave:

Herramientas como JIRA y el tablero Kanban deben seguir siendo utilizadas, ya que han demostrado ser efectivas para la organización y el control del trabajo. Además, el seguimiento de métricas como la velocidad del equipo debe ser aprovechado para planificar de manera realista y motivar al equipo.

Adaptación flexible de metodologías ágiles:

Dado el contexto del sector público, es recomendable seguir adoptando de manera flexible las metodologías ágiles, utilizando aquellas herramientas que mejor se ajusten a las condiciones y necesidades del equipo, sin perder el enfoque en la mejora continua y la colaboración.

Conclusiones

Impacto positivo en el equipo de trabajo:

La implementación de herramientas ágiles generó un entorno de trabajo más organizado y colaborativo, donde los miembros del equipo pudieron visualizar de manera clara el progreso y la evolución de las tareas. Esto tuvo un efecto motivador significativo, ya que observar mejoras en la productividad y eficiencia fomentó una mayor participación e involucramiento en el proceso.

Adaptación exitosa de herramientas ágiles:

Aunque no fue posible implementar metodologías como Scrum en su totalidad debido a limitaciones contextuales y la reducción del equipo, la adopción de herramientas específicas como el tablero Kanban, el planning poker y las reuniones de retrospectiva permitió optimizar el flujo de trabajo y mejorar la planificación. Estas prácticas sentaron las bases para una gestión más eficiente de los proyectos, ajustándose a las necesidades del equipo y del entorno público.

Participación de los stakeholders y reducción de retrabajos:

A pesar de las dificultades iniciales para captar y mantener el interés de los stakeholders, se logró un mayor involucramiento en comparación con proyectos anteriores. Esto se reflejó en un incremento del feedback recibido y, en consecuencia, en la reducción de



retrabajos. Si bien este sigue siendo un desafío, los avances obtenidos destacan la importancia de su participación en el desarrollo ágil de proyectos.

Desafíos enfrentados y aprendizajes obtenidos:

La reducción del equipo de cinco a tres miembros supuso un reto significativo, que impactó en la posibilidad de adoptar roles específicos dentro de Scrum. Esta situación impulsó al equipo a adoptar herramientas ágiles de manera flexible, logrando resultados positivos a pesar de las limitaciones. Además, el proyecto evidenció la necesidad de fortalecer la recopilación de requerimientos desde el inicio, para evitar suposiciones y trabajar con bases más sólidas.

Mejora continua y motivación colectiva:

La aplicación de ciclos de mejora constante a lo largo del proyecto permitió al equipo refinar el uso de herramientas como JIRA, el tablero Kanban y la planificación. Este proceso de mejora continua no solo optimizó el desempeño del equipo, sino que también fortaleció el sentido de colaboración y el compromiso con el proyecto, consolidando una cultura de trabajo más participativa y motivada.

Referencias

- Hernández Sampieri, R. (2018). Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta. McGraw-Hill Education.
- Kleer. (2013). *Proyectos ágiles con Scrum*. Editorial Kleer.
- Patton, J. (2014). User story mapping: Discover the whole story, build the right product. O'Reilly Media.
- Pichler, R. (2016). Strategize: Product strategy and product roadmap practices for the digital age. Pichler Consulting.
- Sutherland, J. (2014). Scrum: The art of doing twice the work in half the time. Crown Business.
- Torres, L., & Ramírez, J. (2018). Making work visible: Exposing time theft to optimize work & flow. IT Revolution Press.
- Varma, A. (2020). The product book: Cómo convertirse en un product manager. Product School.

Apéndice

Guion de Entrevista Semi-estructurada:

1. Introducción:
 - o Agradecimiento por participar.
 - o Explicación breve del propósito de la entrevista.



- Garantía de confidencialidad.
- Solicitar permiso para grabar la entrevista.
- 2. Contexto General del Equipo:
 - ¿Podrías describir brevemente la estructura del equipo? (Tamaño, roles, jerarquía, etc.)
 - ¿Cómo se relacionan y comunican los miembros del equipo entre sí? (Reuniones, canales de comunicación, herramientas colaborativas)
 - ¿Cómo ha cambiado la dinámica del equipo en los últimos meses?
- 3. Distribución del Trabajo:
 - ¿Cómo se asignan las tareas dentro del equipo?
 - ¿Quién o quiénes toman las decisiones sobre qué tareas deben realizarse y en qué orden?
 - ¿Existen criterios o metodologías específicos para la distribución de tareas?
 - ¿Qué tan flexible es la asignación de tareas cuando surgen cambios o nuevas prioridades?
- 4. Toma de Decisiones:
 - ¿Quién decide qué desarrollos se llevarán a cabo en el proyecto?
 - ¿Qué factores se consideran para priorizar el desarrollo de ciertas funcionalidades o módulos?
 - ¿Cómo se decide el tiempo estimado para completar cada desarrollo?
 - ¿Qué tanto participan los desarrolladores en la toma de decisiones sobre qué y cómo se desarrolla?
- 5. Relación con Otras Áreas:
 - ¿Qué tipo de interacción tiene el equipo de desarrollo con otros departamentos o equipos dentro de la institución?
 - ¿Cómo se manejan las solicitudes de cambios o nuevas funcionalidades provenientes de otros departamentos?
 - ¿Existen reuniones regulares entre el equipo de desarrollo y otros departamentos para coordinar esfuerzos?
- 6. Características Técnicas de los Sistemas:
 - ¿Podrías describir brevemente las características técnicas de los sistemas en los que estás trabajando actualmente?



- o ¿Qué desafíos técnicos han enfrentado recientemente en el desarrollo de estos sistemas?
- o ¿Cómo se manejan las decisiones técnicas importantes? ¿Quién tiene la última palabra?
- 7. Opinión y Satisfacción:
 - o ¿Cómo te sientes respecto a la actual dinámica de trabajo en el equipo?
 - o ¿Qué aspectos consideras que funcionan bien y cuáles podrían mejorar?
 - o ¿Qué cambios sugerirías para mejorar la organización del trabajo y la colaboración en el equipo?
- 8. Cierre:
 - o ¿Hay algo más que te gustaría agregar sobre la dinámica de trabajo o algún tema que no hayamos cubierto?

Agenda de Capacitación

Sesión 1: Introducción a las Metodologías Ágiles y Scrum

1. Introducción a las Metodologías Ágiles

- ¿Qué es la agilidad?
- Principios del Manifiesto Ágil.
- Beneficios de trabajar en un entorno ágil.

2. Conceptos Fundamentales de Scrum

- Definición de Scrum y su aplicación en proyectos.
- Los pilares de Scrum: transparencia, inspección y adaptación.
- Roles en Scrum:
 - o Scrum Master: facilitador y coach del equipo.
 - o Product Owner: encargado del backlog y del valor del producto.
 - o Development Team: equipo multidisciplinario encargado de entregar el trabajo.

3. Artefactos de Scrum

- Product Backlog, Sprint Backlog e Incremento.
- Historias de usuario y cómo priorizarlas.



- Concepto de Definition of Done.
- 4. Eventos de Scrum
 - Sprint Planning.
 - Daily Scrum (reunión diaria).
 - Sprint Review y Retrospective.
- 5. Actividad Práctica
 - Implementación del Sprint 0:
 - o Crear un backlog sencillo para el proyecto piloto.
 - o Realizar la primera reunión de planificación.

Sesión 2: Kanban

1. Introducción a Kanban
 - Principios básicos.
 - El flujo de trabajo visual.
2. Elementos de un Tablero Kanban
 - Columnas y tarjetas.
 - WIP (Work In Progress): gestión de límites de trabajo en curso.
3. Scrum y Kanban
 - Integración de ambas metodologías (Scrumban).

Materiales y Recursos

- Presentaciones: Diapositivas con los conceptos clave de Scrum y Kanban.
- Plantillas: Ejemplo de tablero Kanban y backlog de producto.
- Herramientas digitales:
 - o JIRA.
- Fuentes para la capacitación:
 - o Jeff Sutherland - Scrum: The Art of Doing Twice the Work in Half the Time"

Agenda de Reunión con Stakeholders

Cronograma y Agenda (Duración total: 1 hora)

1. Introducción y bienvenida (5 minutos)

Objetivo: Dar contexto general sobre la reunión y el estado actual del proyecto.



Información a obtener: Crear una atmósfera participativa, presentar el objetivo principal de la

reunión y alinear expectativas.

Preguntas: - ¿Están familiarizados con el estado actual del desarrollo? - ¿Hay alguna expectativa específica que tengan antes de comenzar?

2. Presentación de la aplicación (10 minutos)

Objetivo: Mostrar a los stakeholders las funcionalidades actuales de la aplicación.

Información a obtener: Asegurar que comprendan las características claves, así como el propósito

de la aplicación.

Preguntas: - ¿La aplicación cubre las necesidades que ustedes imaginaban? - ¿Qué funcionalidades creen que son más críticas para su trabajo?

3. Prueba de la aplicación en tiempo real (20 minutos)

Objetivo: Permitir que los stakeholders interactúen con la aplicación y prueben sus funcionalidades.

Información a obtener: Feedback sobre la usabilidad, posibles problemas y las características más

valoradas.

Preguntas: - ¿La aplicación es intuitiva y fácil de usar? - ¿Encontraron algún aspecto confuso o difícil de entender? - ¿Qué funcionalidades les parecen más útiles o qué mejoras sugerirían?

4. Recopilación de feedback (15 minutos)

Objetivo: Escuchar sus opiniones, sugerencias y áreas de mejora.

Información a obtener: Comentarios sobre las funcionalidades existentes, sugerencias de nuevas

características y puntos de mejora.

Preguntas: - ¿Qué funcionalidades creen que faltan en la aplicación? - ¿Qué aspectos les gustaría que se mejoraran? - ¿Qué tan bien creen que la aplicación se adapta a las necesidades de su equipo/departamento?

5. Requerimientos para futuras iteraciones (5 minutos)

Objetivo: Identificar las prioridades para los próximos desarrollos.

Información a obtener: Establecer las principales áreas de enfoque para los futuros sprints.



Preguntas: - ¿Cuáles creen que deberían ser las prioridades de desarrollo en las próximas versiones? - ¿Qué funcionalidades adicionales consideran críticas para el éxito de la aplicación?

6. Cierre y próximos pasos (5 minutos)

Objetivo: Resumir lo discutido y definir los siguientes pasos.

Información a obtener: Asegurar que todos los participantes estén alineados con los próximos

pasos y las acciones a tomar.

Preguntas: - ¿Hay alguna otra observación que les gustaría compartir antes de concluir? - ¿Tienen alguna sugerencia adicional sobre cómo deberíamos proceder con las próximas

iteraciones?

Resumen de los objetivos de la reunión:

1. Mostrar la aplicación para asegurar que los stakeholders comprendan su estado actual.
2. Recibir feedback directo sobre la funcionalidad y usabilidad.
3. Identificar mejoras y nuevas funcionalidades que se priorizarán en futuros sprints.
4. Alinear expectativas sobre los próximos pasos y desarrollo futuro de la aplicación.

Entrevista a miembros del equipo de desarrollo post-implementación:

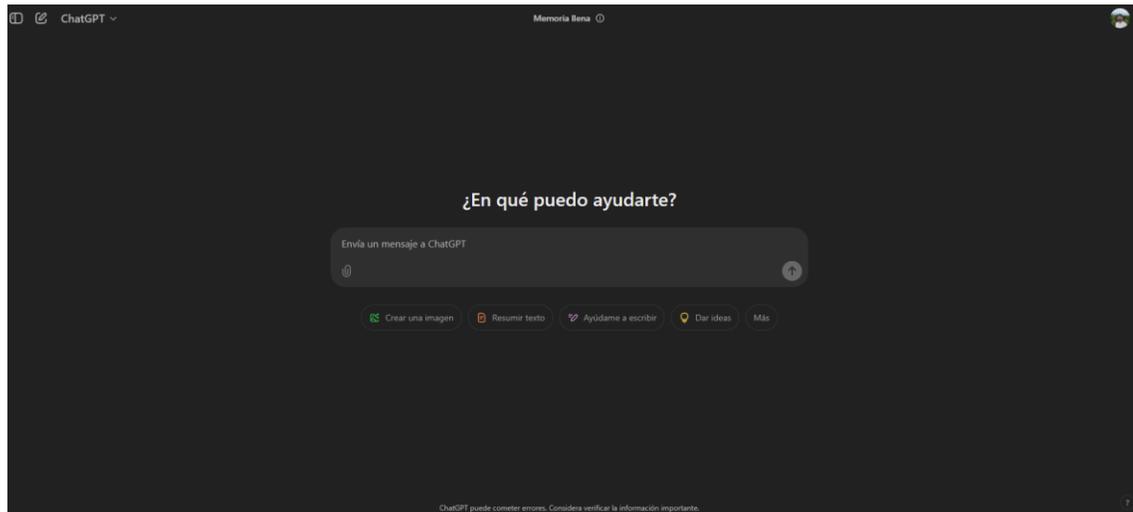
1. ¿Cómo describirías tu experiencia trabajando con metodologías ágiles en este proyecto?
2. ¿Qué cambios notaste en la organización y distribución de las tareas desde que implementamos estas metodologías?
3. ¿Sentís que el uso de herramientas como el tablero Kanban ayudó a mejorar la claridad y el seguimiento de las tareas?
4. ¿Cómo evaluarías la comunicación dentro del equipo antes y después de implementar las metodologías ágiles?
5. ¿Hubo algún aspecto del proceso que te haya resultado más motivador o que te haya dado mayor satisfacción en tu trabajo diario?
6. ¿Considerás que las reuniones de planificación y retrospectiva fueron útiles para ajustar el trabajo y resolver problemas?
7. ¿Qué opinión tenés sobre el nivel de participación e involucramiento de los stakeholders en este proyecto?
8. ¿Hay algo que te gustaría mejorar o cambiar en la forma en que estamos aplicando las metodologías ágiles?



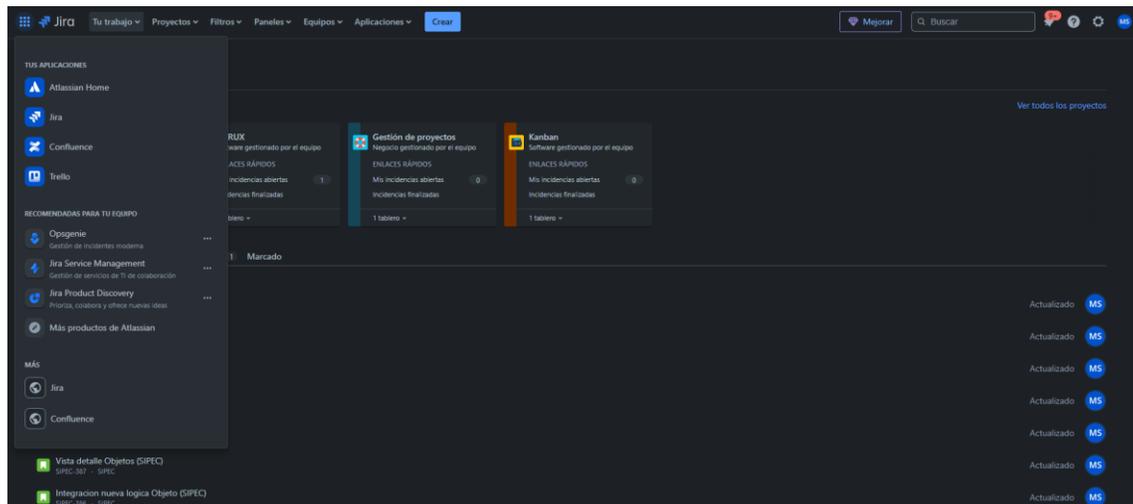
9. ¿Sentís que los resultados obtenidos hasta ahora reflejan una mejora en la eficiencia del equipo? ¿Por qué?
10. En general, ¿recomendarías seguir utilizando metodologías ágiles en futuros proyectos? ¿Por qué?

Anexo

ChatGPT



JIRA





Sección de reportes de JIRA

Resumen

Gráfica de trabajo hecho

Visualiza un trabajo de sprint finalizado y compáralo con su alcance global. Utiliza esta información para llevar a cabo un seguimiento del progreso hasta la finalización del sprint.

Diagrama de trabajo pendiente de sprint

Haz un seguimiento y gestiona el trabajo total restante de un sprint. Una vez finalizado el sprint, resume el rendimiento del equipo y el individual.

Información de velocidad

Puedes predecir la cantidad de trabajo que tu equipo es capaz de asumir para futuros sprints consultando y revisando las cantidades entregadas en los anteriores.

Diagrama de flujo acumulado

Muestra los estados de las incidencias de tu proyecto en el tiempo. Consulta cuáles son las columnas que acumulan más incidencias e identifica cuellos de botella en tu flujo de trabajo.

Informe de duración del ciclo

Descubre cuánto tiempo lleva lanzar incidencias a través de la canalización de implementación y cómo lidiar con los valores atípicos.

Informe de frecuencia de implementación

Conoce tu frecuencia de implementación para conocer los riesgos y la frecuencia con la que lanzas valiosos productos a tus clientes.

Ejemplo de Incidencia en Jira

[SIPEC-281] Crear un Abordaje creado: 02/sep/24 Actualizada: 25/sep/24 Resuelta: 06/sep/24

Estado:	Terminado
Proyecto:	SIPEC
Componentes:	Ninguno
Versiones afectadas:	Ninguno
Versiones corregidas:	Ninguno
Principal:	Intervenciones

Tipo:	Historia	Prioridad:	Medium
Informador:	Desarrollo Tecnológico ECIF	Persona asignada:	Facundo Sanchez
Resolución:	Listo	Votos:	0
Etiquetas:	Ninguno		
Trabajo restante estimado:	Desconocido	Trabajo restante estimado:	Desconocido
Tiempo empleado:	Desconocido	Tiempo Trabajado:	Desconocido
Estimación original:	Desconocido	Estimación original:	Desconocido

Subtareas:	Clave	Resumen	Tipo	Estado	Persona asignada
	SIPEC-293	Crear o revisar el modelo de abordaje...	Subtarea	Terminado	Facundo Sanchez
	SIPEC-294	Generar los serializadores y vistas p...	Subtarea	Terminado	Facundo Sanchez
	SIPEC-295	Generar la sección en el front para l...	Subtarea	Terminado	javier salas.991
	SIPEC-296	Crear el modal con el formulario para...	Subtarea	Terminado	Martin Salas

Sprint: Intervenciones Sprint - 1
Story point estimate: 8
Rank: 0j001y5

Descripción

- Como: Coordinador
- Quiero: Crear un abordaje
- Para: Iniciar el registro del trabajo de campo

Puntos a tener en cuenta para esto

ubicación texto (para el formulario de creación)

no esta definido tipo abordaje (lista de opciones)

preservación del lugar (no esta definido si es una lista)

clima no esta definidas las categorías (consultar lista)