

UNIVERSIDAD CONTINENTAL
MAESTRÍA EN INGENIERÍA DE SOFTWARE
ESCUELA DE POSGRADO



**Universidad
Continental**

PROYECTO ACADÉMICO COLABORATIVO

Análisis de un Caso Práctico en la Gestión de Equipos de Desarrollo de Software

ASIGNATURA : Habilidades y actitudes del Ingeniero de Software.

DOCENTE : Mag. CORDOVA SOLIS, Miguel Angel.

INTEGRANTES:

- AGUILA RUIZ, Bengy Jammil.
- CORILLA MIRANDA, Jersson Jorge.
- QUIÑONES HERMOSA, Luis Paolo.
- VARGAS PEREZ, Franco Mauricio.

FECHA DE ENTREGA : 09 de marzo del 2025

LIMA – PERÚ

2025

Índice

1. Introducción	4
2. Objetivos.....	5
2.1. Objetivo General	5
2.2. Objetivos Específicos.....	5
3. Análisis de Caso Práctico	6
3.1. Descripción del Caso	6
3.1.1. <i>Problemas de Liderazgo en la Integración de un Nuevo Desarrollador.....</i>	<i>6</i>
3.1.2. <i>Resistencia al Cambio.....</i>	<i>6</i>
3.1.3. <i>Deficiencias en la Comunicación del Rol y Expectativas.....</i>	<i>7</i>
3.1.4. <i>Exclusión en la Toma de Decisiones.</i>	<i>7</i>
3.2. Principios Éticos Involucrados.....	8
3.2.1. <i>Principio de Justicia e Igualdad.</i>	<i>8</i>
3.2.2. <i>Principio de Respeto y Dignidad.</i>	<i>8</i>
3.2.3. <i>Principio de Responsabilidad y Transparencia.....</i>	<i>9</i>
3.2.4. <i>Principio de Colaboración y Trabajo en Equipo.</i>	<i>9</i>
3.2.5. <i>Principio de Desarrollo Profesional y Aprendizaje.</i>	<i>10</i>
3.3. Propuesta de Solución.....	10
3.3.1. <i>Reuniones de Integración y Comunicación Abierta.</i>	<i>10</i>
3.3.2. <i>Fomento del Trabajo en Equipo a través del Desarrollo en Pares.....</i>	<i>11</i>
3.3.3. <i>Definición Clara de Roles y Responsabilidades.....</i>	<i>11</i>
3.3.4. <i>Implementación de un Programa de Mentoría.....</i>	<i>12</i>
4. Trabajo Colaborativo en Equipo	12
4.1. Organización interna del equipo y distribución de roles.....	14
4.1.1. <i>Liderazgo y Roles de Scrum.....</i>	<i>14</i>
4.1.2. <i>Composición del Equipo de Desarrollo.....</i>	<i>15</i>
4.1.3. <i>Organización Interna y Ritmo de Trabajo.....</i>	<i>15</i>
4.1.4. <i>Impacto del Enfoque Ágil en la Productividad.....</i>	<i>16</i>
4.2. Estrategias para la toma de decisiones.....	16
4.2.1. <i>Habilidades Blandas en la Toma de Decisiones.....</i>	<i>17</i>
4.2.2. <i>Métricas Clave de Rendimiento (KPIs).</i>	<i>17</i>
4.2.3. <i>Establecimiento de Objetivos y Metas Claras.</i>	<i>18</i>

4.2.4.	<i>Comunicación Efectiva.</i>	18
4.2.5.	<i>Retroalimentación Constructiva.</i>	19
4.3.	Dinámica de trabajo y coordinación de reuniones.	19
4.3.1.	<i>Comunicación Clara y Asertiva.</i>	20
4.3.2.	<i>Trabajo Colaborativo y Adaptabilidad.</i>	20
4.3.3.	<i>Coordinación de Reuniones Efectivas.</i>	20
4.3.4.	<i>Inteligencia Emocional en la Gestión de Conflictos.</i>	21
4.3.5.	<i>Motivación y Reconocimiento.</i>	21
5.	Reflexión Sobre Habilidades Interpersonales	22
5.1.	Importancia en la Gestión de Proyectos	22
5.2.	Experiencia y Decisiones Difíciles	22
5.3.	Acciones para Mejorar la Colaboración y el Liderazgo	22
6.	Resultado del Caso a Resolver	23
7.	Conclusiones y Recomendaciones	24
7.1.	Conclusiones	24
7.2.	Recomendaciones	24
8.	Bibliografía	25

1. Introducción

En la actualidad, la gestión en el desarrollo de software se ha convertido en un campo esencial para el éxito de las organizaciones dedicadas a proyectos tecnológicos, los líderes de proyectos se enfrentan a una serie de desafíos que van más allá de las meras consideraciones técnicas, el presente trabajo explora algunos dilemas éticos y los problemas de liderazgo que emergen en el contexto del desarrollo de software.

El factor humano es considerado el factor más importante en el desarrollo de soluciones de software, muchos líderes de proyectos coinciden en que el triunfo de sus proyectos se debe a las personas y al trabajo de equipo (*Guerrero, L. 2016*), su naturaleza esencialmente colaborativa, implica que las decisiones tomadas por los líderes no solo afectan el resultado del proyecto, sino también la dinámica y el bienestar del equipo de trabajo.

Los dilemas éticos, como la presión por cumplir plazos a expensas de la calidad o la falta de transparencia en la gestión, pueden generar tensiones y afectar la moral del equipo, de la misma forma, los problemas de liderazgo, que incluyen desde la falta de comunicación hasta el manejo inadecuado de conflictos, pueden obstaculizar el avance y limitar el crecimiento del equipo.

Este documento tiene por objetivo analizar un caso concreto, proponiendo estrategias para abordar este tipo de problemas, promoviendo un entorno de trabajo colaborativo y eficiente.

El enfoque del presente trabajo se centra en exponer cómo la gestión efectiva en el desarrollo de software no solo implica cumplir con los objetivos técnicos, sino también cultivar una cultura ética y un liderazgo que fomente el crecimiento tanto personal como profesional dentro del equipo.

2. Objetivos

2.1. Objetivo General

- Analizar un caso práctico relacionado con la gestión de equipos de desarrollo de software, identificando dilemas éticos y problemas de liderazgo para proponer soluciones efectivas alineadas con principios de ética, liderazgo y responsabilidad social.

2.2. Objetivos Específicos

- **Identificar y analizar un caso práctico** en la gestión de equipos de desarrollo de software en el que se presenten dilemas éticos o problemas de liderazgo.
- **Examinar los principios éticos involucrados** en el caso, evaluando su impacto en la toma de decisiones y en la dinámica del equipo.
- **Desarrollar una propuesta de solución ética y efectiva**, considerando enfoques de liderazgo y gestión responsable de equipos.
- **Fomentar el trabajo colaborativo en equipos multidisciplinarios**, promoviendo la toma de decisiones éticas y la resolución de problemas en el desarrollo de software.
- **Reflexionar sobre las habilidades interpersonales y la gestión de proyectos**, identificando áreas de mejora para fortalecer la comunicación, la resolución de conflictos y el liderazgo en entornos profesionales.

3. Análisis de Caso Práctico

3.1. Descripción del Caso

Nuestra empresa se encuentra desarrollando un software integral para la gestión académica, administrativa y financiera de una institución educativa superior. Este proyecto es clave para la organización, ya que busca optimizar procesos como la gestión de matrículas, administración de pagos, generación de reportes académicos y comunicación entre docentes y estudiantes.

El equipo de desarrollo está conformado por dos programadores experimentados, quienes han trabajado juntos durante varios años en múltiples proyectos. Su colaboración ha generado un ambiente de comunicación eficiente, confianza mutua y una cultura de trabajo bien establecida, lo que ha permitido un flujo de trabajo ágil y organizado.

3.1.1. Problemas de Liderazgo en la Integración de un Nuevo Desarrollador.

La llegada de un nuevo programador al equipo de desarrollo generó **tensiones y conflictos internos**, afectando la dinámica del grupo y la eficiencia del proyecto. Los miembros más experimentados, acostumbrados a tomar decisiones y liderar el proceso, comenzaron a **percibir la presencia del nuevo integrante como un desafío a su autoridad**. Esta situación creó resistencia al cambio, deterioró el ambiente laboral y afectó la integración del nuevo colaborador.

3.1.2. Resistencia al Cambio.

Uno de los principales desafíos fue la **resistencia del equipo senior** a aceptar la incorporación del nuevo programador. Al considerar que su experiencia y conocimiento eran superiores, algunos miembros adoptaron una postura excluyente, generando un ambiente de

trabajo hostil. Como resultado, el nuevo programador **se sintió poco bienvenido, temeroso de hacer preguntas y reacio a proponer sugerencias**, lo que limitó su aprendizaje y desarrollo profesional.

Según Kotter (2012), la resistencia al cambio en los equipos de trabajo suele estar vinculada al miedo a la pérdida de control, a la incertidumbre y a la falta de comunicación efectiva. En este caso, la falta de una estrategia clara para la integración del nuevo programador exacerbó estos factores, dificultando su adaptación.

3.1.3. Deficiencias en la Comunicación del Rol y Expectativas.

El líder del equipo no estableció desde el inicio **una comunicación clara sobre el rol del nuevo programador ni su contribución dentro del equipo**. La ausencia de directrices precisas generó confusión y malentendidos, afectando la colaboración y el sentido de pertenencia del nuevo integrante.

De acuerdo con Goleman, Boyatzis y McKee (2013), un liderazgo efectivo requiere empatía y comunicación clara para fomentar un entorno de confianza y trabajo en equipo. Sin una orientación adecuada, es común que los equipos establezcan barreras internas que impidan la integración de nuevos miembros.

3.1.4. Exclusión en la Toma de Decisiones.

Cuando el nuevo programador se unió al equipo, los desarrolladores más antiguos **continuaron tomando decisiones sin incluirlo en el proceso**, lo que reforzó su sensación de

exclusión. Esta falta de participación no solo limitó su aprendizaje, sino que también afectó su motivación y compromiso con el proyecto.

La teoría del liderazgo transformacional propuesta por **Bass y Riggio (2006)** enfatiza que los líderes deben inspirar y guiar a su equipo mediante la inclusión y el desarrollo del talento. Un liderazgo efectivo debe promover la participación activa de todos los miembros y evitar estructuras jerárquicas que excluyan a ciertos individuos del proceso de toma de decisiones.

3.2. Principios Éticos Involucrados

3.2.1. Principio de Justicia e Igualdad.

El principio de justicia establece que todas las personas deben ser tratadas de manera equitativa y sin discriminación (**Beauchamp & Childress, 2019**). En este caso, el nuevo programador no recibe un trato igualitario, ya que es excluido de la toma de decisiones y del proceso de trabajo. La falta de equidad afecta su integración y desarrollo profesional, lo que representa una vulneración a este principio.

Relevancia:

Garantizar un trato justo fomenta la colaboración y la eficiencia del equipo, asegurando que todos los miembros tengan las mismas oportunidades de participación y crecimiento.

3.2.2. Principio de Respeto y Dignidad.

Todo miembro de un equipo merece ser tratado con respeto y dignidad, independientemente de su nivel de experiencia. La hostilidad y exclusión que enfrenta el

programador junior afectan su autoestima y bienestar emocional. Según Goleman, Boyatzis y McKee (2013), un entorno de trabajo tóxico reduce la productividad y desmotiva a los empleados.

Relevancia:

El respeto es clave para crear un ambiente de trabajo saludable donde todos los colaboradores se sientan valorados y motivados para contribuir con sus habilidades y conocimientos.

3.2.3. Principio de Responsabilidad y Transparencia.

El líder del equipo tiene la responsabilidad de garantizar una comunicación clara y un ambiente de trabajo armonioso. La falta de transparencia en la definición de roles y expectativas generó confusión y conflictos en el equipo. Según Bass y Riggio (2006), un líder efectivo debe actuar con integridad y comunicar claramente los objetivos del equipo para evitar malentendidos.

Relevancia:

La transparencia en la gestión del equipo permite mejorar la confianza y la cohesión grupal, facilitando la resolución de conflictos y la toma de decisiones éticas.

3.2.4. Principio de Colaboración y Trabajo en Equipo.

El trabajo en equipo es esencial en el desarrollo de software. La resistencia de los programadores experimentados a incluir al nuevo miembro obstaculiza la cooperación y el intercambio de conocimientos. La ética profesional exige fomentar la colaboración y evitar conductas que generen exclusión o competencia desleal (IEEE Code of Ethics, 2020).

Relevancia:

Promover una cultura de colaboración aumenta la productividad y permite que el equipo aproveche mejor sus talentos y experiencias individuales.

3.2.5. Principio de Desarrollo Profesional y Aprendizaje.

El ingreso de un programador junior debería representar una oportunidad para que el equipo **comparta conocimientos y fomente el aprendizaje continuo**. Sin embargo, al no permitirle participar activamente, se limita su desarrollo profesional y su crecimiento dentro de la empresa.

Relevancia:

El desarrollo profesional continuo es clave para la innovación y mejora en los proyectos. Un equipo que comparte conocimientos y experiencias es más eficiente y adaptable a los cambios tecnológicos.

3.3. Propuesta de Solución

Reconociendo los conflictos de liderazgo y comunicación dentro del equipo, el líder decidió implementar diversas estrategias para fomentar un ambiente de trabajo más colaborativo, inclusivo y productivo. Estas estrategias se enfocaron en mejorar la integración del nuevo programador, fortalecer la comunicación y garantizar un liderazgo efectivo.

3.3.1. Reuniones de Integración y Comunicación Abierta.

Se establecieron reuniones semanales donde cada miembro del equipo tenía la oportunidad de expresar sus ideas, preocupaciones y sugerencias. Esto permitió abrir canales de comunicación,

fomentar la transparencia y generar confianza entre los desarrolladores (**Goleman, Boyatzis & McKee, 2013**).

Justificación: La comunicación efectiva es fundamental en equipos de desarrollo de software, ya que facilita la resolución de conflictos y mejora la cohesión del grupo (**Bass & Riggio, 2006**).

3.3.2. Fomento del Trabajo en Equipo a través del Desarrollo en Pares.

Se implementó la metodología de **programación en pares (pair programming)**, donde el nuevo programador trabajaba directamente con un desarrollador experimentado. Este enfoque permitió fortalecer la colaboración, reducir errores y facilitar el aprendizaje mutuo (**Williams & Kessler, 2002**).

Justificación: La programación en pares ha demostrado ser efectiva en la mejora de la productividad y en la integración de nuevos desarrolladores, al proporcionarles un entorno de aprendizaje guiado y reducir su curva de adaptación (**Cockburn & Highsmith, 2001**).

3.3.3. Definición Clara de Roles y Responsabilidades.

El líder del equipo estableció y comunicó claramente los roles y responsabilidades de cada miembro, asegurando que el nuevo programador tuviera tareas específicas que pudiera realizar con supervisión. Esto le permitió ganar confianza y aprender gradualmente dentro del equipo (**Northouse, 2018**).

Justificación: Una estructura organizativa clara mejora la eficiencia y reduce la incertidumbre en la distribución de tareas, lo que fortalece la moral del equipo y evita conflictos internos (Yukl, 2013).

3.3.4. Implementación de un Programa de Mentoría.

Para facilitar la adaptación del nuevo programador, se asignó un mentor senior que lo guiara en su proceso de integración. La mentoría brindó un espacio seguro para resolver dudas, recibir retroalimentación y fomentar un sentido de pertenencia dentro del equipo (Scandura & Williams, 2004).

Justificación: La mentoría en el entorno laboral contribuye significativamente al desarrollo de habilidades técnicas y a la integración organizacional de los nuevos empleados (Allen, Eby, Poteet, Lentz, & Lima, 2004).

4. Trabajo Colaborativo en Equipo

En la gestión de proyectos de software basada en Scrum, el trabajo colaborativo es esencial para el éxito y la eficiencia del equipo. Scrum es una metodología ágil que promueve la colaboración estrecha entre los miembros del equipo para adaptarse rápidamente a los cambios y entregar productos de alta calidad.

- **Importancia del Trabajo Colaborativo en Scrum**

Scrum se fundamenta en la idea de que los equipos multifuncionales y autoorganizados pueden gestionar proyectos complejos de manera más efectiva. La colaboración permite que cada miembro aporte sus habilidades y conocimientos específicos, enriqueciendo el proceso de

desarrollo y fomentando soluciones innovadoras. Además, la comunicación constante y abierta mejora la transparencia y la confianza dentro del equipo.

- **Resultados del Trabajo Colaborativo en Scrum**

La implementación de un enfoque colaborativo en Scrum conlleva múltiples beneficios:

- **Mayor Adaptabilidad:** La colaboración facilita la respuesta rápida a los cambios en los requisitos o en el entorno del proyecto, permitiendo ajustes oportunos en el desarrollo.
- **Incremento de la Productividad:** La sinergia entre los miembros del equipo optimiza el uso de recursos y tiempo, mejorando la eficiencia en la ejecución de tareas.
- **Mejora en la Calidad del Producto:** La diversidad de perspectivas y la revisión conjunta de avances contribuyen a identificar y corregir errores de manera temprana, elevando la calidad del software entregado.

- **Desafíos Potenciales del Trabajo Colaborativo en Scrum**

A pesar de sus beneficios, la colaboración en equipos Scrum puede enfrentar ciertos desafíos:

- **Problemas de Comunicación:** Diferencias en la interpretación de información o falta de claridad pueden generar malentendidos que afecten el progreso del proyecto.
- **Resistencia al Cambio:** Algunos miembros pueden mostrar reticencia a adoptar nuevas prácticas o herramientas colaborativas, lo que puede limitar la efectividad del equipo.

- **Gestión de Conflictos:** La interacción constante puede dar lugar a desacuerdos o tensiones que, si no se abordan adecuadamente, pueden afectar la cohesión del equipo.

Para mitigar estos desafíos, es fundamental fomentar una cultura de comunicación abierta, habilidades blandas, establecer roles y responsabilidades claras, y promover la formación continua en habilidades técnicas y blandas. De esta manera, el trabajo colaborativo en Scrum se convierte en un pilar sólido para la gestión exitosa de proyectos de software.

4.1. Organización interna del equipo y distribución de roles.

El equipo de desarrollo opera bajo la metodología **Scrum**, una de las más utilizadas en entornos ágiles debido a su capacidad para fomentar la colaboración, la adaptabilidad y la entrega incremental de valor (**Schwaber & Sutherland, 2020**). Dentro de esta estructura, cada miembro desempeña un **rol clave** que contribuye a la eficiencia y el éxito del proyecto.

4.1.1. Liderazgo y Roles de Scrum.

El liderazgo del equipo está a cargo de **Jersson Jorge Corilla Miranda**, quien asume un **doble rol**:

- **Scrum Owner:** Responsable de la visión del producto y la gestión del **Product Backlog**, asegurando que el equipo priorice las tareas con mayor impacto y alineación con los objetivos del proyecto (**Rubin, 2012**).
- **Scrum Master:** Facilita la comunicación, elimina impedimentos que puedan afectar el flujo de trabajo y promueve el cumplimiento de las prácticas ágiles, garantizando que

el equipo siga el marco de trabajo de **Scrum** de manera efectiva (**Sutherland & Schwaber, 2017**).

4.1.2. Composición del Equipo de Desarrollo.

El equipo está conformado por **tres programadores experimentados** y un nuevo integrante, con funciones claramente definidas:

- **Bengy Jammil Águila Ruiz, Franco Vargas Pérez y Luis Quiñones Hermosa** se encargan de la implementación de las soluciones técnicas, asegurando la calidad del código y el cumplimiento de los requerimientos.
- **Luis Quiñones Hermosa**, el nuevo programador, se encuentra en proceso de adaptación al equipo y a la metodología de trabajo. Recibe soporte continuo para integrarse rápidamente y contribuir de manera efectiva al desarrollo del producto.

Según la **Guía Scrum (Schwaber & Sutherland, 2020)**, los equipos de desarrollo son **autoorganizados y multifuncionales**, lo que les permite trabajar de manera autónoma en la entrega de incrementos de producto de alta calidad.

4.1.3. Organización Interna y Ritmo de Trabajo.

La estructura interna del equipo se basa en la **colaboración y la transparencia**, fundamentales en metodologías ágiles (**Cockburn & Highsmith, 2001**). Para garantizar una sincronización efectiva y la mejora continua, se realizan las siguientes reuniones clave dentro de cada iteración o **Sprint**:

- **Daily Stand-ups:** Reuniones diarias cortas donde los desarrolladores comparten avances, identifican bloqueos y alinean esfuerzos.
- **Sprint Planning:** Sesión de planificación donde se definen las tareas a desarrollar en el Sprint.
- **Sprint Review:** Presentación del incremento de producto desarrollado al final de cada Sprint.
- **Sprint Retrospective:** Reflexión sobre el desempeño del equipo para identificar áreas de mejora y optimizar procesos.

4.1.4. Impacto del Enfoque Ágil en la Productividad.

La aplicación rigurosa de Scrum permite que el equipo funcione de manera eficiente, asegurando que los **objetivos del proyecto se cumplan con calidad y dentro de los tiempos establecidos (Rubin, 2012)**. Además, el uso de un marco ágil estructurado promueve la adaptabilidad a cambios, la optimización del trabajo en equipo y la entrega continua de valor al cliente **(Highsmith, 2009)**.

4.2. Estrategias para la toma de decisiones.

En scrum la toma de decisiones se basa en el trabajo del equipo, donde se combinan habilidades blandas, datos claros, objetivos definidos, comunicación abierta y retroalimentación útil.

4.2.1. Habilidades Blandas en la Toma de Decisiones.

Las **habilidades blandas** juegan un papel fundamental en la promoción de un entorno de trabajo colaborativo, facilitando la comunicación efectiva, la toma de decisiones basadas en información y el consenso dentro del equipo de desarrollo. En el contexto de **Scrum**, estas habilidades permiten una interacción armoniosa entre los miembros del equipo, impulsando la productividad y el bienestar organizacional (**Goleman, 1998**).

- **Empatía:** Comprender y valorar las perspectivas del equipo fortalece la confianza y el trabajo en conjunto (**Neff, 2011**).
- **Colaboración:** Fomentar la participación activa mejora la creatividad y el compromiso con los objetivos del equipo (**Lencioni, 2002**).
- **Resolución de Conflictos:** Manejar desacuerdos de forma constructiva permite transformar conflictos en oportunidades de mejora (**Fisher, Ury & Patton, 2011**).
- **Liderazgo Servicial:** El Scrum Master y el Product Owner deben actuar como facilitadores, guiando al equipo sin imponer decisiones (**Greenleaf, 2002**).

4.2.2. Métricas Clave de Rendimiento (KPIs).

Las métricas brindan información objetiva para mejorar la toma de decisiones y optimizar el desempeño del equipo.

- **Velocidad del Equipo:** Mide el trabajo completado por sprint, permitiendo predecir la capacidad en futuras iteraciones.
- **Tasa de Finalización (Burn-down Chart):** Visualiza el progreso del sprint y facilita ajustes en la planificación.

- **Calidad del Producto:** Indicadores como bugs reportados y satisfacción del cliente ayudan a priorizar mejoras.
- **Retroalimentación de Stakeholders:** Evalúa la percepción del usuario final para optimizar entregas y prioridades.

4.2.3. Establecimiento de Objetivos y Metas Claras.

En scrum la toma de decisiones efectivas es importante que el equipo tenga claro los objetivos y metas.

- **Objetivos del Sprint (Sprint Goals):** Definidos durante la planificación del sprint, estos objetivos guían al equipo en la selección de tareas y en la toma de decisiones diarias.
- **Visión del Producto:** El Product Owner debe comunicar claramente la visión del producto y cómo cada sprint contribuye a ella.
- **Criterios de Aceptación:** Cada tarea o historia de usuario debe tener criterios claros que definan cuándo está "Terminada" (Definition of Done).

4.2.4. Comunicación Efectiva.

La comunicación es la base para la toma de decisiones en equipo. En Scrum, se promueve a través de:

- **Reuniones Estructuradas:** Las reuniones diarias (Daily Scrum), las revisiones de sprint (Sprint Review) y las retrospectivas (Sprint Retrospective) aseguran que todos estén alineados y tengan la información necesaria.

- **Transparencia:** El tablero de Scrum (Scrum Board) y el backlog deben ser visibles y accesibles para todos.
- **Escucha Activa:** Todos los miembros del equipo deben escuchar y respetar las opiniones de los demás, fomentando un ambiente de confianza.

4.2.5. Retroalimentación Constructiva.

La retroalimentación es una herramienta potente para optimizar el proceso de toma de decisiones. En Scrum, se fomenta a través de:

- **Sprint Retrospective:** Al final de cada sprint, el equipo reflexiona sobre lo que funcionó y lo que no, identificando áreas de mejora.
- **Feedback Continuo:** Durante las revisiones de sprint, los stakeholders proporcionan feedback sobre el trabajo completado, lo que ayuda a ajustar prioridades.
- **Cultura de Mejora Continua:** El equipo debe estar abierto a recibir y dar feedback de manera constructiva, sin culpas ni juicios.

4.3. Dinámica de trabajo y coordinación de reuniones.

La organización interna del equipo y las estrategias de toma de decisiones son fundamentales para establecer una dinámica de trabajo eficiente y una coordinación efectiva en un entorno Scrum. Según **Schwaber y Sutherland (2020)**, Scrum se basa en la transparencia, inspección y adaptación, lo que requiere una estructura clara y procesos bien definidos.

4.3.1. Comunicación Clara y Asertiva.

La estructura del equipo, liderado por Jersson Jorge Corilla Miranda en los roles de Scrum Owner y Scrum Master, promueve una comunicación abierta y transparente. **Según Schwaber y Sutherland (2020)**, la comunicación efectiva en Scrum es clave para garantizar la alineación del equipo y la resolución rápida de impedimentos. Durante las reuniones diarias (*Daily Stand-ups*), cada miembro, incluyendo a los programadores Bengy Jammil Águila Ruiz, Franco Vargas Pérez y Luis Quiñones Hermosa, comparte sus avances, próximos pasos e impedimentos. Esta práctica fortalece la confianza y la cohesión del equipo **(Rubin, 2012)**.

4.3.2. Trabajo Colaborativo y Adaptabilidad.

En las sesiones de *Sprint Planning*, el equipo colabora para definir los objetivos del sprint y asignar tareas según las habilidades y cargas de trabajo de cada miembro. Esta colaboración se basa en la confianza mutua y en la disposición para adaptarse a cambios, elementos clave en la metodología ágil **(Pichler, 2019)**. La diversidad de perspectivas y experiencias enriquece la toma de decisiones y la resolución de problemas, alineándose con los principios de mejora continua en Scrum **(Schwaber & Sutherland, 2020)**.

4.3.3. Coordinación de Reuniones Efectivas.

Las reuniones en Scrum están diseñadas para ser eficientes y productivas. Cada ceremonia tiene un propósito claro: las *Sprint Reviews* permiten presentar el trabajo completado y recibir retroalimentación de los *stakeholders*, mientras que las *Sprint Retrospectives* ofrecen un espacio para reflexionar sobre el proceso y proponer mejoras **(Rubin, 2012)**. La participación activa y el

respeto por los tiempos establecidos son fundamentales para el éxito de estas reuniones (Schwaber & Sutherland, 2020).

4.3.4. Inteligencia Emocional en la Gestión de Conflictos.

El liderazgo de Jersson Jorge Corilla Miranda como *Scrum Master* es crucial para manejar conflictos de manera constructiva. **Según Goleman (1998)**, la inteligencia emocional desempeña un papel fundamental en la gestión de equipos de trabajo, permitiendo reconocer y gestionar emociones propias y ajenas para facilitar interacciones armoniosas. Fomentar un ambiente donde las diferencias se aborden con empatía y se busquen soluciones colaborativas fortalece la dinámica del equipo y previene la escalada de tensiones (**Pichler, 2019**).

4.3.5. Motivación y Reconocimiento.

Reconocer los logros individuales y colectivos es esencial para mantener la motivación. La retroalimentación positiva y la celebración de éxitos refuerzan el compromiso del equipo y la satisfacción en el trabajo (**Pink, 2011**). Además, el apoyo al nuevo integrante, Luis Quiñones Hermosa, en su proceso de adaptación, refleja el valor que se le da al desarrollo y bienestar de cada miembro del equipo, lo que contribuye a un entorno de aprendizaje continuo y colaboración efectiva (**Rubin, 2012**).

Integrando estos aspectos de organización interna y estrategias de toma de decisiones, el equipo Scrum establece una dinámica de trabajo sólida y una coordinación de reuniones efectiva, pilares fundamentales para el éxito en la gestión de proyectos ágiles.

5. Reflexión Sobre Habilidades Interpersonales

5.1. Importancia en la Gestión de Proyectos

En entornos ágiles como Scrum, las habilidades interpersonales son fundamentales para el éxito del equipo y la entrega eficiente del producto. La comunicación clara, la empatía y el liderazgo colaborativo permiten una mejor coordinación, reducen conflictos y fomentan un ambiente de trabajo positivo (*Schwaber & Sutherland, 2020*). En el equipo liderado por Jersson Jorge Corilla Miranda, estas habilidades han sido esenciales para garantizar que cada miembro pueda aportar al máximo de su capacidad, asegurando que el nuevo programador, Luis Quiñones Hermosa, se integre sin afectar el rendimiento general.

5.2. Experiencia y Decisiones Difíciles

Uno de los desafíos más significativos fue la incorporación de un nuevo miembro al equipo. La falta de experiencia en la metodología Scrum y la necesidad de adaptación al ritmo de trabajo presentaron obstáculos que pudieron haber afectado la productividad y la moral del equipo. Para abordar esta situación, se implementaron estrategias como la mentoría personalizada, el trabajo en pares y la asignación progresiva de tareas según el nivel de experiencia. Estas decisiones no solo facilitaron su integración, sino que también fortalecieron el sentido de pertenencia dentro del equipo, alineándose con las mejores prácticas ágiles en la gestión de talento (*Pichler, 2019*).

5.3. Acciones para Mejorar la Colaboración y el Liderazgo

Para garantizar una cultura de trabajo eficiente y colaborativa, se han adoptado acciones clave, como reuniones semanales de integración para compartir inquietudes y fortalecer la cohesión del equipo. Además, el liderazgo servicial del Scrum Master ha sido fundamental para gestionar conflictos y facilitar el desarrollo profesional de cada miembro. Al fomentar un ambiente

de confianza y comunicación abierta, se logra que el equipo sea más resiliente ante los cambios y que los proyectos avancen con mayor fluidez (*Rubin, 2012*).

6. Resultado del Caso a Resolver

Con el tiempo, las tensiones iniciales en el equipo fueron disminuyendo a medida que el nuevo programador, Luis Quiñones Hermosa, se adaptaba al entorno de trabajo ágil. Gracias a una estrategia de mentoría personalizada y sesiones de trabajo en pares, logró no solo integrarse sin dificultades, sino también aportar nuevas ideas y enfoques innovadores que enriquecieron el desarrollo del proyecto.

A su vez, los programadores senior comenzaron a reconocer su potencial y habilidades, lo que fortaleció la colaboración y el aprendizaje mutuo dentro del equipo. Este proceso permitió mejorar la sinergia y la confianza entre los miembros, optimizando la eficiencia en la ejecución de tareas y favoreciendo la adopción de nuevas prácticas ágiles.

El liderazgo servicial del Scrum Master fue clave en esta transformación, al proporcionar un entorno donde la retroalimentación constructiva y el aprendizaje continuo se convirtieron en pilares del trabajo en equipo. Como resultado, la dinámica general del grupo mejoró significativamente, aumentando la productividad y reforzando el compromiso de todos con los objetivos del proyecto.

7. Conclusiones y Recomendaciones

7.1. Conclusiones

7.2. Recomendaciones

8. Bibliografía

Allen, T. D., Eby, L. T., Poteet, M. L., Lentz, E., & Lima, L. (2004). Career benefits associated with mentoring for protégés: A meta-analysis. *Journal of Applied Psychology*, 89(1), 127-136. <https://doi.org/10.1037/0021-9010.89.1.127>

Bass, B. M., & Riggio, R. E. (2006). *Transformational leadership* (2nd ed.). Lawrence Erlbaum Associates.

Beauchamp, T. L., & Childress, J. F. (2019). *Principles of biomedical ethics* (8th ed.). Oxford University Press.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer*, 34(11), 131-133. <https://doi.org/10.1109/2.963450>

Fisher, R., Ury, W., & Patton, B. (2011). *Getting to yes: Negotiating agreement without giving in*. Penguin Books.

Guerrero, L. P. C. (2016). Gestión en proyectos de software. *Tecnología Investigación y Academia*, 4(2), 12-19.

Goleman, D. (1998). *Working with emotional intelligence*. Bantam Books.

Goleman, D., Boyatzis, R., & McKee, A. (2013). *Primal leadership: Unleashing the power of emotional intelligence*. Harvard Business Review Press.

Greenleaf, R. K. (2002). *Servant leadership*. Paulist Press.

Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley.

IEEE. (2020). IEEE Code of Ethics. Retrieved from <https://www.ieee.org/about/corporate/governance/p7-8.html>

- Kotter, J. P. (2012). *Leading change*. Harvard Business Review Press.
- Lencioni, P. (2002). *The five dysfunctions of a team*. Jossey-Bass.
- Neff, K. (2011). *Self-compassion*. HarperCollins.
- Northouse, P. G. (2018). *Leadership: Theory and practice* (8th ed.). SAGE Publications.
- Pichler, R. (2019). *How to lead in product management: Practices to align stakeholders, guide development teams, and create value together*. Pichler Consulting.
- Pink, D. H. (2011). *Drive: The surprising truth about what motivates us*. Riverhead Books.
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular agile process*. Addison-Wesley.
- Scandura, T. A., & Williams, E. A. (2004). Mentoring and transformational leadership: The role of supervisory career mentoring. *Journal of Vocational Behavior*, 65(3), 448-468.
<https://doi.org/10.1016/j.jvb.2003.10.003>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.
- Sutherland, J., & Schwaber, K. (2017). *Scrum: The art of doing twice the work in half the time*. Crown Business.