

Tarea de mejora al proceso 5. Reflexión de Aplicación de Lean Software Development

Principio	% inicial	% actual	Explicacion	Mejoras
Eliminar desperdicios	30	80	Conforme avanzó el desarrollo, se aplicaron estrategias como la automatización de cálculos, normalización de la base de datos y una estructura modular de endpoints para evitar retrabajos. Además, se mejoró la comunicación interna mediante reuniones de revisión semanal, lo cual redujo los errores por malentendidos.	También se establecieron lineamientos de documentación en los endpoints y en la estructura de datos, evitando confusiones futuras.
Amplificar aprendizaje	20	65	Dentro de este tiempo, se cree que se ha mejorado este principio por las diferentes experiencias que cada integrante del grupo ha tenido, algunos más que otros, pero, dentro de este proyecto ciertos integrantes han logrado trabajar en otras áreas del proyecto para expandir los conocimientos, junto con los problemas que supimos resolver en lo que se avanzaba el proyecto como tal, como la refactorización y el deployment. No se cumplen todas las	Las mejoras en este principio se lograron mediante la rotación de integrantes del equipo a nuevas áreas del proyecto, lo cual fomentó la expansión de conocimientos. Además, la resolución grupal de problemas técnicos complejos, específicamente durante la refactorización y el <i>deployment</i> , generó un valioso conocimiento práctico que fue compartido entre los miembros.

			técnicas, pero si las suficientes como para subir de porcentaje.	
Tomar decisiones lo mas tarde posible	50	75	Se puede ver el resultado de este principio dentro de las gráficas burndown de jira, ya que hemos hecho muchas más reuniones para discutir que es lo que se puede hacer en debidos avances del proyecto, especialmente para cuando se refactorizaron partes de este mismo, aunque si se mantuvieron las opciones abiertas, también si le garantizamos al product owner que podíamos hacer lo que fuera agregando al proyecto, aunque interfiriera con el proceso actual. Y para poder apegarnos a las fechas para trabajar se tenían que dejar planeadas las tareas de manera rápida, lo cual interfirió con este principio como tal.	En este aspecto, las mejoras se materializaron a través de un aumento en la frecuencia de las reuniones de discusión y planificación. Esto permitió una toma de decisiones más informada, flexible y consensuada, especialmente al abordar tareas críticas como la refactorización del código, y ayudó a mantener las opciones abiertas antes de comprometerse con una solución definitiva.
Entregar lo antes posible	60	85	Esto fuimos cumpliendo cada vez más en lo que avanzaba el proyecto ya que, esencialmente el proyecto se trata de un sistema de inventario, y todo lo que se le iba agregando se iba construyendo a base de eso, el movimiento de materiales depende completamente de como este implementado el inventario, y así durante todo el proyecto, no completamos el principio ya que faltan ciertos aspectos para que sea completamente funcional, pero el product owner ha	La mejora principal aquí fue la implementación exitosa de una estrategia de entrega incremental. Se priorizó la entrega de la funcionalidad base (el inventario) para luego construir los módulos dependientes sobre esa base ya funcional. Este enfoque facilitó la validación y aprobación continua por parte del <i>Product Owner</i> en cada avance presentado.

			estado contento con todo lo que hemos hecho y a aprobado en los avances que hemos hechos.	
Potenciar al equipo	75	80	Junto con tomar decisiones lo más tarde posible, se puede notar que empezábamos a organizarnos más, por ende, tomar en cuenta lo que otro integrante del grupo decía, para así facilitar a la hora planear y distribuir las tareas, también entre cada sub grupo (back y front) se hacia el esfuerzo de poder organizarse para dejar en un tiempo apropiado las tareas así hay más conexión para poder llegar al producto final en las diferentes etapas del proyecto, también comunicándonos para saber cómo es que cada integrante va hacer las cosas.	Manifestaron en un notable incremento de la organización general del equipo. Se empezó a tomar más en cuenta la opinión de cada integrante, lo cual facilitó directamente la planificación y distribución de las tareas. Hubo un esfuerzo consciente de coordinación entre los sub-grupos para alinear sus tiempos y entregas, generando una mayor conexión para alcanzar el producto final en cada etapa. Esto se reforzó con una comunicación más activa sobre cómo cada miembro planeaba ejecutar sus responsabilidades.
Crear la integridad	40	90	Durante el desarrollo, la integridad del sistema fue una prioridad, especialmente por la interdependencia entre módulos como inventario, proyectos y usuarios. Se integraron validaciones cruzadas en el backend (por ejemplo, evitar salidas de materiales sin stock suficiente) y se implementaron relaciones bien definidas en la base de datos en 3FN. También se aplicó una lógica de auditoría para rastrear movimientos, reforzando la confiabilidad del sistema.	Se mejoró la integridad aplicando pruebas de integración y endpoints con control de errores más robustos.

Visualizar todo el conjunto	80	95	En las primeras etapas, el equipo tendía a trabajar en partes separadas (por ejemplo, inventario y proyectos) sin considerar el impacto global. Con el avance del curso, la visión se amplió al entender cómo cada módulo influye en el flujo general del negocio	Actualmente, el equipo trabaja con una visión integral del sistema. Se utilizan diagramas actualizados y documentación visual para planificar nuevas funcionalidades considerando la arquitectura completa. La mejora principal es mantener esa visión global en futuras fases, incluyendo mantenimiento y posibles integraciones con otros sistemas administrativos.
-----------------------------	----	----	---	---

Conclusiones

Durante el curso, el equipo ha pasado de aplicar los principios de diseño de manera parcial a una integración mucho más sólida y consciente, alcanzando un promedio general cercano al 90 %. Esto se refleja en la madurez del sistema, que ahora cuenta con módulos interconectados, una base de datos optimizada y flujos de trabajo más claros y eficientes.

El equipo alcanzó una visión más integral del proyecto, comprendiendo cómo cada componente contribuye al valor total del producto. Esto les permitió diseñar soluciones más escalables y alineadas con los objetivos del product owner.

Al principio nuestro trabajo era más desorganizado y con tareas repetitivas, pero con el tiempo aprendimos a planificar mejor, refactorizar código y mantener una comunicación más clara entre los miembros del equipo.