

**UNIVERSIDAD DEL VALLE DE GUATEMALA**

Ingeniería de Software 2, sección 30



## **TAREA DE MEJORA AL PROCESO**

Sofia Garcia - 22210

Julio Garcia Salas - 22076

Joaquin Campos - 22155

Juan Fernando Menéndez - 17444

Juan Pablo Solis - 22102

**GUATEMALA, Noviembre de 2024**

## Reflexión sobre la Aplicación de Principios Lean en el Proyecto DivinoSeas

**Tabla de Principios Lean**

Principio	Porcentaje	Explicación	Mejoras
Eliminar Desperdicios	40%	Hemos identificado redundancia en algunas funciones de la API y variables que ya no se usan.	Eliminar código redundante, variables no usadas, y mejorar la revisión de código para minimizar desperdicios.
Amplificar el Aprendizaje	70%	Implementamos sesiones de revisión de código semanales y programación en pareja para mejorar habilidades técnicas.	Incrementar la frecuencia de sesiones de intercambio de conocimientos, incorporar un Wiki de documentación.
Tomar Decisiones lo Más Tarde Posible	60%	Retrasamos la decisión de tecnologías hasta evaluar todas las opciones. Ha mejorado nuestra precisión en decisiones.	Documentar casos donde retrasar decisiones fue beneficioso, y mejorar la planificación para decisiones futuras.
Entregar lo Antes Posible	50%	Hemos reducido el tiempo entre entregas, priorizando historias de usuario claves.	Implementar un proceso de retroalimentación con stakeholders para ajustar la frecuencia de entregas.
Potenciar al Equipo	75%	Los desarrolladores participan en decisiones de tiempos y priorización, aumentando su	Crear sesiones formales para feedback del equipo y ajustar responsabilidades en función del feedback recibido.

		autonomía y confianza.	
Crear la Integridad	60%	Contamos con integración continua, pero las pruebas automatizadas aún no cubren todo el sistema.	Aumentar la cobertura de pruebas automatizadas y mejorar el pipeline de builds para garantizar estabilidad.
Visualizar Todo el Conjunto	55%	Hemos mejorado nuestra capacidad para ver el proyecto como un todo, no solo en partes individuales.	Implementar reuniones de revisión global para alinear mejor la visión del proyecto en su conjunto.

## Reflexión Comparativa y Respuesta a Interrogantes

### Eliminación de Desperdicios

- Desperdicios Identificados: Redundancia en funciones de la API y variables sin uso.
- Reducción Lograda: Mediante revisiones periódicas de código.
- Estrategias Eficaces: Revisión de código y retroalimentación constante entre el equipo.
- Requisitos Poco Claros: Se establecieron criterios claros desde el inicio, y la comunicación frecuente con stakeholders ha reducido cambios innecesarios.

### Amplificación del Aprendizaje

- Métodos Implementados: Programación en pareja y revisiones de código.
- Cambio en Habilidad para Compartir Conocimientos: Mayor confianza entre los miembros y aumento en la frecuencia de las sesiones de aprendizaje.
- Áreas Mejoradas: Habilidades en manejo de la API y buenas prácticas de documentación.

### Tomar Decisiones lo Más Tarde Posible

- Decisiones Retrasadas: Selección de herramientas y tecnologías hasta contar con requisitos claros.
- Desafíos: A veces, el retraso en decisiones técnicas generó incertidumbre. Mejora: crear un cronograma para decisiones clave.

### Entrega lo Antes Posible

- Reducción del Tiempo Entre Entregas: Sí, ha beneficiado la agilidad del proyecto, aunque aún puede mejorarse.

- Retroalimentación Recibida: Los usuarios agradecen entregas más rápidas, lo que permite ajustar el producto de acuerdo con sus necesidades.

## **Potenciar al Equipo**

- Promoción de la Participación: Integración del equipo en decisiones de tiempo y tareas, aumentando la autonomía.
- Desafíos en Autonomía vs. Objetivos: A veces, se priorizó la autonomía sobre los plazos. Mejora: balancear autonomía y cumplimiento de objetivos.

## **Crear la Integridad**

- Mejoras en Integración Continua: Se incrementó la automatización, pero la cobertura de pruebas sigue en progreso.
- Dificultades en Mantener la Integridad: Complejidad en builds. Mejora: optimizar el proceso de integración.

## **Visualizar Todo el Conjunto**

- Cambio en la Visión del Proyecto: Ahora el equipo es más consciente del impacto global.
- Mejoras: Sesiones de alineación para analizar el proyecto como un sistema interconectado.

## **Conclusiones Generales**

En general, nuestro equipo ha logrado aplicar un promedio del 55% de los principios de Lean Software Development en el proyecto DivinoSeas. A lo largo del curso, hemos avanzado en la implementación de prácticas más eficientes y en la reducción de desperdicios, mejorando notablemente desde la evaluación inicial en julio. Este proceso nos ha permitido entender mejor los beneficios de un enfoque Lean y cómo aplicarlo efectivamente en el desarrollo de software. Desde julio, nuestro equipo ha incrementado la aplicación de los principios en aproximadamente un 20%, con esfuerzos especiales en áreas de aprendizaje continuo, integración de equipo y visualización global del proyecto.

## **Ejemplos de Eliminación de Desperdicios**

Un ejemplo concreto de la reducción de desperdicios fue la eliminación de funciones redundantes en nuestra API y variables que no se utilizaban en el código, lo cual mejoró el rendimiento y la claridad del sistema. Además, establecimos una revisión de código semanal para identificar y reducir errores y elementos no usados en el proyecto. Desde entonces, hemos mejorado la eficiencia del código y disminuido la cantidad de refactorización necesaria.

## **Ejemplos de Amplificación del Aprendizaje**

Implementamos programación en pareja y sesiones de revisión de código con una frecuencia semanal. Esto nos permitió identificar errores y mejorar las habilidades del equipo en el manejo de la API de forma colaborativa. Por ejemplo, uno de los miembros

desarrolló un enfoque más optimizado para la autenticación en la API, que luego compartió con el resto del equipo. Estas sesiones incrementaron significativamente la capacidad de compartir conocimientos y reducir errores en las funciones de autenticación.

## **Ejemplos de Tomar Decisiones lo Más Tarde Posible**

Un caso notable de aplicación de este principio fue la selección de herramientas para la gestión de despliegue. En lugar de comprometernos desde el inicio, investigamos varias opciones de despliegue hasta que estuvimos seguros de nuestras necesidades. Esto nos ayudó a evitar cambios de último minuto y mejoró nuestra capacidad de toma de decisiones informada.

## **Ejemplos de Entrega lo Antes Posible**

Para reducir el tiempo entre entregas, definimos prioridades claras en nuestras historias de usuario, enfocándonos en entregar primero las funcionalidades más solicitadas por los usuarios. Por ejemplo, la función de gestión de inventarios fue implementada y entregada antes de las demás debido a su alto impacto en las operaciones del client