

Plan de Acción de Retrospectiva (Abril-Junio)

Este documento resume las iniciativas de mejora surgidas de la retrospectiva del equipo. Cada punto de acción incluye una descripción del problema a resolver, su prioridad y los criterios que definen cuándo se considerará completado.

Los participantes fueron Orlando, Dany, Daniel, Adrian y Omar

01: Establecer un Proceso de Desarrollo de Software Estandarizado

- Owner: Dany Peñaloza
- **Prioridad:** P0

Múltiples comentarios de la retrospectiva señalaron la necesidad de un flujo de trabajo más claro y robusto. Esto incluye la gestión de código con Git, el proceso de revisión de Pull Requests (PRs), la estrategia de pruebas y los despliegues. Actualmente, la falta de un estándar provoca inconsistencias, dificulta la integración de código y aumenta el riesgo de introducir errores en producción. Este punto de acción centraliza varios feedbacks en una única iniciativa para crear un ciclo de vida de desarrollo predecible y de alta calidad.

- **Criterios de Aceptación (Definition of Done):**
 - ☐ Se ha definido y documentado un flujo de branching en Git (ej. GitFlow o similar) que especifica cómo y cuándo se crean y fusionan las ramas `feature`, `release`, y `main/master`.
 - ☐ Se ha creado una plantilla de Pull Request que incluye secciones para el contexto del cambio, el enlace al ticket correspondiente y un checklist de auto-revisión para el autor.
 - ☐ Se define e implementa una estrategia de pruebas que clarifica qué se espera de las pruebas unitarias, de integración y los smoke tests.
 - ☐ El proceso de release y deployment está documentado, especificando los pasos y responsables para mover código entre los entornos de Staging y Producción.
 - ☐ Se realiza una sesión con todo el equipo para presentar y explicar el nuevo proceso estandarizado.

02: Implementar un Marco de Gestión de Producto y Proyecto

- Owner: Angel (PM) / Nicole (PO)
- **Prioridad:** P0

El equipo manifestó una desconexión con la visión del producto y la necesidad de un mejor manejo de las prioridades ("Todo es prioridad"). A menudo se comienzan tareas sin el contexto adecuado, lo que lleva a retrabajo. Esta iniciativa busca implementar un marco de trabajo ágil más estructurado e involucrar a Product/Business para asegurar que el equipo trabaje en las funcionalidades de mayor valor y con los requerimientos claros desde el inicio.

- **Criterios de Aceptación (Definition of Done):**

- ☐ Se establece un ciclo de feedback formal y recurrente con el equipo de Producto para revisar necesidades y alinear expectativas.
- ☐ Se adopta un sistema claro para la definición de tareas, donde cada historia de usuario debe incluir Criterios de Aceptación claros y/o un "Definition of Done" (DoD) acordado con Producto antes de iniciar el desarrollo.
- ☐ Se genera y mantiene un roadmap de producto visible para el equipo, que define los objetivos a corto y mediano plazo (trimestral).
- ☐ Se implementan ceremonias ágiles clave (ej. Sprint Planning, Daily Scrums por escrito o breves, Sprint Review/Demo) para mejorar la comunicación y el seguimiento del progreso.
- ☐ El equipo de desarrollo tiene la potestad de rechazar o solicitar más información sobre tareas que no cumplan con los criterios de definición mínimos.

03: Crear un Proceso de Onboarding y Colaboración para Desarrolladores

- Owner: Orlando
- **Prioridad:** P1

Se identificó la necesidad de contar con un proceso definido tanto para la incorporación de nuevos miembros al equipo como para la colaboración con desarrolladores externos y/o remotos. La falta de un onboarding estructurado retrasa la curva de aprendizaje y la productividad de los nuevos integrantes o complica la coordinación de desarrolladores remotos.

- **Criterios de Aceptación (Definition of Done):**

- ☐ Se ha creado un checklist de onboarding que incluye:
 - Accesos necesarios a repositorios, herramientas y sistemas.
 - Guía de configuración del ambiente de desarrollo local.
 - Documentación clave a revisar (arquitectura, procesos, guías de estilo).
 - Una primera tarea simple ("good first issue") para familiarizarse con el flujo.
- ☐ Se ha definido y documentado un flujo de trabajo para la colaboración con equipos externos y/o remotos, especificando canales de comunicación, puntos de contacto y

expectativas de entrega, responsabilidades esperadas.

04: Mejorar la Visibilidad del Progreso a través de Demos y Comunicación Asíncrona

- Owner: Daniel
- **Prioridad:** P1

Para mejorar el alineamiento y el ciclo de feedback, el equipo necesita mostrar su trabajo de manera más efectiva. Las sesiones de "Show, don't tell" (demos) permiten obtener retroalimentación temprana y celebrar los logros. Complementariamente, se necesita una forma eficiente de comunicar avances de forma escrita para mantener informados a todos los stakeholders.

- **Criterios de Aceptación (Definition of Done):**
 - ☐ Se establece una sesión de Demo al final de cada ciclo de desarrollo (ej. quincenal) donde el equipo muestra el trabajo completado a los stakeholders.
 - ☐ Se define una plantilla o formato simple para reportar avances de forma asíncrona (ej. un breve resumen semanal en un canal de Slack) para comunicar progreso y bloqueos.

05: Optimizar los Entornos de Desarrollo

- Owner: Dany
- **Prioridad:** P1
- **Descripción:** El feedback señaló que el ambiente de desarrollo está "hardcodeado" y no es representativo de producción, lo que dificulta las pruebas y puede causar sorpresas en los despliegues.
- **Criterios de Aceptación (Definition of Done):**
 - ☐ Se realiza un plan para que el ambiente de Staging sea un reflejo fiel de Producción, definiendo los clientes y escenarios que debe soportar.
 - ☐ Se documenta un proceso para mantener el ambiente de Staging limpio y funcional para las pruebas.

06: Explorar Herramientas de IA

- Owner: Orlando
- **Prioridad:** P1

Hubo interés en explorar herramientas de IA para mejorar la productividad. Esta iniciativa se enfoca en mejorar el tooling y los entornos de trabajo, mejorar el producto con el uso de AI no

entra en el scope de esta tarea.

- **Criterios de Aceptación (Definition of Done):**
 - ☐ Investiga y presenta una o dos herramientas de IA (ej. linters con IA, asistentes de código) que puedan ser incorporadas en el flujo de trabajo, junto con una recomendación sobre su adopción.
 - ☐ Definir un estrategia para mejorar el prompting de la empresa con ejemplos de su uso, así como transmitir conocimiento sobre conceptos de AI (ej. modelos, agentes, prompting, etc).

Appendice A: Feedback

Start Doing

que Omar no se fresee y entre a las reuniones	4
Mejorar el flujo de git , releases y deployments	4
Tener un flujo/onboarding definido para nuevos desarrolladores	4
Estar presente en toda la arquitectura nueva que se esta implementando	4
Mas revision de pr's	4
invitarme a los summits, sobre todo a los de cankun o los cabos	4
Alineamiento con el producto, me parece que hizo falta	4
Revision de PR's	4
Retar más las capacidades del equipo	3
Involucrar a todo el equipo en las ceremonias del sprint	3
mejorar las pruebas de unidad, reggression tests y smoke test	3
revision de PR mas a consciencia, que traigan contexto, ticket, dejar mas preguntas	3
Ciclo de retroalimentacion con producto, cuales son las necesidades, que esta funcionando bien y que no esta funcionando bien	3
Mejor ambiente de desarrollo que sea representativo de produccion	3
Generar o pedir con producto / negocio una ruta de hacia donda vamos	3
limpiar es estado de staging, definir los clientes y escenarios que debemos soportar y poder hacer pruebas	3
Buscar maneras de incorporar IA en el trabajo diario, tratando de que nos ahorre tiempo en ciertas actividades	3

que Omar no se fresee y entre a las reuniones	4
el manejo del proyecto, puede verse beneficiado en ciclos mas cortos, y comunicacion mas eficiente, dailys por escrito, burn downs, estimaciones, grooming, velocity, backlog, sprint planning	3
mas comunicaciones entre el equipo, intenciones y necesidades, incluso desde la parte directiva	3
Feedback del refactor, deberia ser mas interactivo, el codigo no es mio ya que se integra si no del equipo	2
la relacion con desarrolladores externos puede mejorarse	2
diseñar y planear mas la solucion, y validarla con alguien mas antes de comenzar a programarla	2
Roadmap a largo plazo, mediano y corto plazo, con definicion por trimestre que es el objetivo y las tareas para complirlo (tanto como quieran meterse incluso llegando a OKRs)	2

Stop Doing

Decir que si a todo, empezar a limitar o filtrar	4
Agregar codigo a las carreras para completar con cierto tiempo de entrega	4
Aceptar tareas sin contexto adecuado	4
Todo es prioridad 🔥	4
Comenzar a codear antes de conocer el problema a resolver	3
Reaccionar a las situaciones y mejor ser proactivo a los problemas que salgan por cambios	3
Aprobar PR sin entender que se esta agregando o los efectos de los cambios	3
Quedarse callado cuando hay dudas	3
Que la validacion de las features pase solo por nosotros, deberia estar involucrado otro equipo	3
ambiente de desarrollo super hardcodeado	3
Generar toda la informacion de un task desde el punto de vista de ingenieria sin tomar en cuenta a los demas stakeholders	2

Continue doing it

Comunicacion clara sobre los avances del proyecto	4
Creando las historias de manera tan detallada como se viene haciendo desde hace un par de sprints.	4
features que dan valor al producto tentativamente (double ledger entry, v2 refacor)	4
Seguir con la buena actitud a pesar de los problemas que se puedan presentar o molestias	4
Seguir teniendo estas reuniones que se dan de vez en cuando acerca de temas de mejora (GIT - DDD, ETC)	4
configurar linters para normalizar codigo, pre-commit, husky, incluso agregar linters IA	3
Detallar los issues	3
creacion de un canal con comunicacion directa conmigo	1