

Feedback primera entrega 2023: Larenas, Nahuel (260079), Barreto, Piñeyro, Sofía (243498), Varela, Florencia (254005). 15/04/23.

1. Comentarios generales sobre el informe.
 - a. Muy buena estructura.
 - b. Fácil de leer.
2. Sobre la Pequeña guía de "Definición/uso del proceso de ingeniería en el contexto de KANBAN".
 - a. Lo denominaron "Guía del marco Kanban para el proyecto". Traten de mantener la misma denominación que se propone en el obligatorio ("Definición/uso del proceso de ingeniería en el contexto de KANBAN". Ídem para todos los otros artefactos en todas las entregas). Esto facilita la búsqueda en el repositorio.
 - b. Les faltó una columna: "Que resultados se obtiene de cada fila".
 - c. En la columna codificación y pruebas debieron tener en cuenta en el "Cómo" a TDD. Es fundamental.
 - d. En los roles ponen a los testers, pero no los nombran nunca en el "Quién". O los sacan de la última tabla o los ponen en el "Quién".
 - e. Esta definición la contrastamos contra el tablero de la entrega 2.
 - f. Lo que no queda claro qué es la columna "Integración". No está ni este doc. ni el doc. de la segunda entrega "Explicación del tablero". Deberían aclarar para que la usen en el doc. correspondiente de la tercera entrega. Esto tómelo como una lección aprendida.
3. La Pequeña guía de "Creación y posterior mantenimiento del repositorio: elementos que contiene y cómo los van a versionar".
 - a. Está correcta y clara.
 - b. Quizás convenga justificar mejor la estrategia de branching: ¿por qué GitFlow y no trunkbase? ¿Qué ventajas les da una frente a la otra? ¿Por qué se descarta trunkbase?
4. Sobre el "Análisis de deuda técnica".
 - a. Completo.
 - b. Muy buen uso de herramientas.
 - c. Muy buena la estrategia para testear y encontrar problemas de codificación.
 - d. Para reportar los bugs se utilizó un único criterio facilitando el entendimiento de cada uno de los errores reportados.
 - e. Todos los issues fueron correctamente clasificados y priorizados.
 - f. Quizás convenga incluir una tabla resumen con los issues detectados, de forma tal de poder tener una mirada general de qué tantos inconvenientes se tienen en el proyecto y que tan alta/baja es la deuda técnica.
 - g. Dado que todo el proyecto se está trabajando en español. se recomienda que los issues también sean reportados en español.
 - h. Las issues que se detectaron con los analizadores estáticos de código carecen de descripción.
 - i. Hubiera sido bueno comentar un poco más sobre las reglas seleccionadas para el análisis estático de código, así como también presentar un resumen de todo lo detectado.

5. El “Detalle de registro de esfuerzo por tipo de tarea, según las actividades del proceso de ingeniería o del tablero, si se prefiere” y los “Totales de registro de esfuerzo por la entrega.”
 - a. Están muy bien.
 - b. En issues deberían poner fecha de inicio, fecha de fin y el esfuerzo de cada una.
 - c. De todas maneras, deberían de alguna forma relacionarlo con las fechas de inicio y fin los totales de las tareas.
6. Sobre el “Informe de la retrospectiva.”
 - a. Los post-its están bastante ilegibles. Muy chiquitos.
 - b. Algunas de las acciones planteadas aparecen como muy confusas y ambiguas. Por ej.: ¿Qué quiere decir “Coordinar reuniones muy sobre la fecha”? ¿Cuáles reuniones, qué fecha?
Otro ejemplo: “Comunicación de trabajo en equipo”: ¿Cuál trabajo? ¿A quién comunicar? ¿Cuándo comunicar?
Revisen los post-its que estén ambiguos.
Algunas cosas las pudimos inferir leyendo las conclusiones, pero ese no es el objetivo de las conclusiones.
 - c. Veo que plantearon algunos problemas de gestión (“Dedicar horas necesarias al proyecto”), pero no pudimos ver problemas de desarrollo o ingeniería o técnicas. Además, nos pareció que les faltan algunas: ¿creen Uds. que están todos los temas importantes para que un equipo reflexione y mejore? Les mencionamos algunas que **pueden** (no estamos afirmando, esto lo deben determinar Uds.) haber omitido: métricas, estimaciones, riesgos, DoD, capacitación, cuellos de botella, entre otros. También pueden agregar otros.
 - d. Antes de llenar los cuatro cuadrantes les sugerimos que utilicen algún método para que no se olviden u omitan lo mencionado anteriormente y que además busquen una manera de asegurarse que el post-it que pongan es útil. Por ej.: preguntarse ¿por qué voy a poner este post-it? ¿Qué problema busco resolver? ¿El problema es de parte del equipo o de todo el equipo? ¿El problema se da siempre o a veces? ¿Es un problema realmente importante? Etc.
 - e. Las conclusiones están correctas y en línea con lo que debería ser una conclusión, pero omite algunos post-its que mencionan en los cuadrantes.
 - f. De la DAKI no solo salen conclusiones sino lecciones aprendidas que las pueden ir anotando en un backlog para ir mejorando en sucesivas entregas y para hacerlas referencias en la entrega final. Pueden salir también dificultades y desafíos. Ídem en un backlog.
 - g. Les dejamos este link que es el método para retrospectivas que vimos el semestre pasado: <https://retromat.org/blog/best-retrospective-for-beginners/>. Si tienen dudas, lo vemos en clase.

Sugerencias:

1. Quizás podrían poner un pequeño Readme apenas se entra al repositorio para que el lector tenga una idea con que se va a encontrar. Con 5 o 6 líneas alcanza.
2. Permitan copiar desde los pdfs que nos obligan a escribir demás.