

Explicación del tablero y su vínculo con el proceso de ingeniería

De acuerdo a lo mencionado en la entrega anterior, el equipo implementará dos tableros Kanban, uno para la gestión y creación de la documentación solicitada en cada entrega y las tareas administrativas asociadas, y el otro para la gestión de tareas de codificación (asociadas al proceso de desarrollo) y creación de los artefactos correspondientes.

Así, modificamos el tablero de gestión anterior para solamente incluir 3 columnas:

1. *Todo*, con las tareas pendientes de ser realizadas;
2. *In Progress*, con las tareas en ejecución; y,
3. *Done*, con las tareas finalizadas.

No se utiliza una columna para la retrospectiva en sí, sino que se utilizan tarjetas que representan los artefactos que deben crearse como resultado de la misma.

El tablero de gestión resultante es un tablero kanban ágil y, pese a solamente tener una columna para aquellas tareas realizándose, sirve para mantener el orden y una visualización rápida del estado administrativo de la entrega.

Por su parte, el tablero Kanban para el desarrollo de código debe tener las mismas columnas que el proceso de ingeniería, pero en esta entrega algunas de ellas se omiten dado que no tenemos ingeniería de requerimientos asociada. De esta forma el tablero resultante es el siguiente: Estas columnas incluyen:

- *ToDo*, representa todas las tareas pendientes para el equipo de desarrollo. Esta columna incluye todas las funcionalidades y requisitos definidos para el proyecto.
- *Test Cases Implementation*, se refiere a la elaboración y definición de los casos de prueba necesarios para verificar que la funcionalidad desarrollada cumpla con los requisitos definidos.
- *Application Implementation*, se centra en la codificación y desarrollo de la funcionalidad en sí misma. Aquí es donde se desarrolla el código, siguiendo las especificaciones y los requisitos definidos previamente.
- *Testing (Automation Tools)*, se enfoca en la automatización de las pruebas, utilizando herramientas específicas para mejorar la eficiencia y la precisión de las pruebas.
- *Refactoring*, se enfoca en la mejora continua del código y su mantenimiento a largo plazo. En esta fase, se realizan mejoras y ajustes al código para mejorar su calidad, rendimiento y escalabilidad.
- *Review*, se utiliza para revisar y validar el trabajo realizado hasta el momento. En esta fase, se revisa el código, los casos de prueba, la documentación y cualquier otro elemento relevante para asegurar que se cumplen los requisitos del proyecto.
- *Done*, columna donde se depositan todos los elementos que finalizan todas las etapas.