

Definición/uso del proceso de ingeniería en el contexto de KANBAN

Para esta entrega, hemos realizado una actualización en el proceso de ingeniería, basado en Behavior Driven Development (BDD) y enmarcada en el contexto de Kanban. En particular, hemos realizado cambios para hacer más claras las actividades que deben realizarse en cada etapa.

El proceso de ingeniería comienza entonces por la etapa de **Definición de Requerimientos**, que corresponde a la actividad denominada anteriormente como BDD feature. En esta actividad, creamos las user stories para las nuevas funcionalidades, utilizando el formato "Como, Quiero, Para", y agregando criterios de aceptación presentados como escenarios de BDD utilizando el formato "Dado, Cuando, Entonces". De esta manera, logramos una mayor comprensión de los requerimientos de cada funcionalidad, lo que nos permite trabajar con mayor eficiencia.

Luego, hemos decidido dividir la actividad de **Codificación y Pruebas Unitarias** en dos actividades separadas: **Pruebas Unitarias** y **Codificación**. Con esta separación, logramos una mayor claridad en las responsabilidades, lo que a su vez facilita el seguimiento y control del proceso de ingeniería en el tablero de GitHub.

En la actividad de **Pruebas Unitarias**, nos enfocaremos exclusivamente en la creación y ejecución de pruebas para verificar el correcto funcionamiento de las funcionalidades desarrolladas. Esto nos permitirá asegurar la calidad del código y detectar posibles errores o fallos antes de proceder con la siguiente etapa.

Por otra parte, en la actividad de **Codificación**, nos centraremos en el desarrollo del código correspondiente a las funcionalidades del proyecto. Aquí, podremos concentrarnos en implementar las soluciones requeridas y cumplir con los requisitos establecidos, contando con la certeza de que las pruebas unitarias previas han validado su correcto funcionamiento.

Posteriormente, hemos tomado la decisión de reemplazar la actividad **Integración** por la actividad **Refactor**. Esta modificación surge debido a que hemos automatizado el proceso de integración, pero nos dimos cuenta de la necesidad de incluir una actividad específica para la refactorización del código.

El objetivo de la actividad **Refactor** es mejorar la estructura, eficiencia y legibilidad del código existente, sin alterar su funcionalidad. Mediante esta práctica, buscamos mantener un código limpio, promoviendo una mejor mantenibilidad.

Luego, hemos agregado la actividad **Tests de integración** con el fin de automatizar la integración. En esta etapa, llevamos a cabo pruebas exploratorias utilizando la herramienta Selenium para asegurarnos de que todas las partes del sistema estén funcionando correctamente.

En la etapa de **revisión del producto**, realizamos una revisión junto con el product owner para verificar que el producto cumpla con los criterios de aceptación. Este proceso es más efectivo gracias a la implementación de BDD, ya que los criterios de aceptación se definen desde el

principio y se verifican a lo largo del proceso. En esta etapa, también se verifica que los tests se ejecuten correctamente.

Por último, en la etapa de **retrospective**, evaluamos el trabajo que se realizó, revisando las métricas y la eficiencia del equipo. Para esto, se utiliza el método DAKI, lo que permite identificar las fortalezas y debilidades del equipo y del proceso de ingeniería actualizado y definir mejoras.