

# Proceso de Ingeniería

El proceso de ingeniería de software se refiere a la planificación, diseño, construcción, prueba y deploy de software. Este proceso se puede dividir en varias fases, cada una de las cuales tiene su propio conjunto de actores, procesos y entregables, y se realizan en un tiempo determinado..

En general, las fases del proceso de ingeniería de software incluyen la identificación de requerimientos, el diseño de la solución, la implementación, la prueba y el mantenimiento del software. Es importante destacar que, en la actualidad, existe una gran variedad de metodologías de desarrollo de software que pueden ayudar a gestionar el proceso de ingeniería de software, y cada una de ellas tiene sus propias ventajas y desventajas, para el proyecto utilizaremos la metodología Kanban, basada en BDD (Behavior Driven Development).

Kanban es una de las metodologías ágiles de desarrollo de software que se utilizan comúnmente en el proceso de ingeniería de software, su principal foco se encuentra en mejorar la eficiencia y la calidad del software mediante la visualización del flujo de trabajo, lo que nos permite detectar defectos del proceso mediante la identificación de cuellos de botella.

BDD es una metodología de desarrollo de software que se enfoca en el comportamiento del sistema desde la perspectiva del usuario final. En lugar de centrarse en las funciones y características técnicas del software, BDD se centra en el comportamiento esperado del software en diferentes situaciones y escenarios.

El proceso de Ingeniería constará de los siguientes pasos:

## 1. Requirement Definition:

- El ingeniero de requerimientos será el encargado de definir los requerimientos del proyecto en forma de historias de usuario, teniendo en cuenta las necesidades del cliente y las restricciones del proyecto.
- Utilizará la técnica de ccc (card conversation confirmation) para asegurarse de que los requerimientos se entienden correctamente y están bien especificados.
- Esta etapa será la primera en realizarse, antes de empezar con cualquier otra actividad del proyecto.
- Al finalizar se obtendrán las historias de usuario, con una especificación clara de lo que se espera lograr con cada una.

## 2. Test Cases Implementation:

- El tester es el encargado de realizarlo.
- Utilizará herramientas y técnicas de prueba apropiadas para implementar y validar cada caso de prueba. El tester se asegurará que los casos de prueba sean completos, precisos y estén bien documentados.
- Esta etapa se realizará después de que se hayan definido las historias de usuario y antes de que comience la fase de desarrollo.

- Al finalizar se tendrán los casos de prueba implementados y listos para ser ejecutados para asegurar la calidad del software desarrollado.
- 3. Application Implementation:
  - Los desarrolladores serán los encargados de codificar la solución definida en la etapa anterior.
  - Utilizarán la técnica de TDD (Test Driven Development) para escribir los tests unitarios y automáticos correspondientes a cada funcionalidad.
  - Esta etapa se llevará a cabo después de que se haya definido la arquitectura completa del proyecto.
  - Al finalizar se obtendrán piezas de software funcionales que cumplan con los requerimientos definidos en la etapa de ingeniería de requerimientos.
- 4. Testing (Automation Tools):
  - Se realizará de forma automática luego de finalizar la codificación.
  - Lo realizará con el framework de tests que corresponda.
  - Esta etapa se llevará a cabo a medida que se vaya liberando código.
  - Al finalizar se obtendrá una lista de issues y/o piezas de software funcional y testeado.
- 5. Refactoring:
  - Los desarrolladores, son los encargados de refactorizar el código que haya pasado las pruebas.
  - Lo realizan basándose fuertemente en los tests unitarios creados anteriormente.
  - Esta etapa se lleva a cabo luego de que los tests automáticos pasaron. Las pruebas automáticas se ejecutan nuevamente y el refactor solamente es aceptable si no introduce comportamientos inesperados.
  - Al finalizar se obtendrán piezas de software funcional, testeado y ordenado.
- 6. Revisión:
  - Es el Product Owner el encargado de la revisión de que se haya logrado lo solicitado por el cliente.
  - Lo realiza en representación del cliente, o en contacto con él.
  - Esta etapa sucede una vez que se tengan piezas de software probadas y funcionales.
  - Al finalizar se obtendrá software que cumple las expectativas o necesidades inicialmente planteadas por el PO.
- 7. Entrega o deploy:
  - Los encargados de DevOps serán los responsables de implementar el software en producción.
  - Esta etapa se llevará a cabo al ejecutar cada commit de la rama principal.
  - Se automatizará mediante herramientas como GitHub Actions.
  - Al finalizar se tendrá software en producción listo para ser utilizado por los usuarios finales.
- 8. Retrospectiva:
  - El equipo completo llevará a cabo una reunión de retrospectiva al final del proyecto.
  - Esta reunión será supervisada por el facilitador de Kanban y se realizará utilizando el método DAKI (Drop, Add, Keep, Improve).
  - Se realizará antes de realizar la entrega final de cada etapa.

- Al finalizar se obtendrá una revisión del trabajo realizado, mejores prácticas, oportunidades de mejora, etc.

## Definición de roles

Del proceso de ingeniería que nos planificamos realizar se extraen los siguientes roles:

- Ingeniero de requisitos (Horacio Abalos): es el encargado de definir los requerimientos del proyecto en forma de historias de usuario, teniendo en cuenta las necesidades del cliente y las restricciones del proyecto. Utiliza la técnica de ccc (card conversation confirmation) para asegurarse de que los requerimientos se entienden correctamente y están bien especificados.
- Desarrollador (Todos): es el encargado de programar la solución diseñada por el arquitecto, utilizando las prácticas de TDD (Desarrollo Guiado por Pruebas) y escribiendo test unitarios y automáticos. Todos los integrantes del equipo actuarán como desarrolladores en todas las instancias del proyecto.
- Tester (Todos): es el encargado de implementar y validar cada caso de prueba, utilizando herramientas y técnicas de prueba apropiadas. Se asegura que los casos de prueba sean completos, precisos y estén bien documentados. Este rol se desempeña después de definir las historias de usuario y antes de la fase de desarrollo.
- Product Owner (Rotativo, inicialmente Horacio Ábalos): es el representante del cliente y quien toma las decisiones sobre el producto final, y decide si las tareas están terminadas. Este rol será ejercido por un estudiante, que será el responsable de coordinar con el cliente para asegurar que el producto final cumpla con los requerimientos.
- DevOps (Denise Souberville): es el encargado de asegurarse de que los procesos de integración y entrega de software sean fluidos y eficientes.
- Facilitador de Kanban (Rotativo): es la persona encargada de asegurar que el flujo de trabajo en el tablero Kanban se esté llevando a cabo de manera correcta y en tiempo y forma. Este rol irá rotando entre los integrantes del equipo en cada instancia.

Es importante mencionar que la rotación de roles en cada instancia permitirá que todos los integrantes del equipo tengan la oportunidad de aprender y desarrollar nuevas habilidades en diferentes áreas del proyecto. Además, todos los miembros del equipo tendrán responsabilidades en la codificación y las pruebas, para garantizar la calidad del software desarrollado.