

# Actividad 5: Estrategia de Control de Calidad y Pruebas Unitarias

*Unax Aller*

15/03/2025

## Estrategia de Control de Calidad y Pruebas Unitarias

El objetivo de esta estrategia es asegurarse de que el software funcione bien y evitar problemas en el futuro haciendo pruebas unitarias y revisando el código. Si se encuentran errores pronto es mucho más fácil arreglarlos y ayuda a que el desarrollo sea más estable y ordenado.

### Enfoque en Pruebas Unitarias

Uso de Frameworks de Testing

Se va a usar JUnit para hacer pruebas unitarias y comprobar que cada clase y método importante funcionen como deberían. Estas pruebas no solo servirán para ver si todo va bien, sino también para ver cómo reacciona el sistema cuando algo falla.

Cobertura Integral

Las pruebas cubrirán tanto casos en los que todo funciona como se espera como situaciones raras para asegurarse de que el sistema responde bien en cualquier momento.

Integración Continua

Las pruebas unitarias se harán automáticamente cada vez que alguien haga un cambio en el código o suba una nueva versión. Para esto se usará un pipeline de integración continua como Jenkins o GitHub Actions. Así se pueden encontrar errores rápido y mantener el código estable.

### Validación del Código y Análisis Estático

Análisis Estático

Se van a usar herramientas como SonarQube, PMD o Checkstyle para revisar el código y encontrar errores, problemas de seguridad y cosas que no sigan las reglas de programación que se han puesto.

Revisión de Código

El equipo mirará el código de vez en cuando para compartir buenas prácticas, encontrar errores lógicos y mejorar la estructura del software entre todos.

### Contribución a la Reducción de Deuda Técnica

Detección Temprana de Errores

Si se usan pruebas unitarias automatizadas y análisis estático, se pueden encontrar problemas desde el principio, lo que ahorra tiempo y esfuerzo en arreglarlos más tarde.

Mantenimiento y Evolución

Tener un código bien probado y documentado hace que sea más fácil añadir cosas nuevas o hacer cambios sin que se rompa nada.

## Desarrollo Guiado por Pruebas

Si se usa TDD, se diseña el software de una manera más ordenada, asegurando que todo funcione bien desde el principio y ayudando a que crezca mejor en el futuro.