

Depuración de errores

Pruebas de caja

Caja blanca

Caja negra

Test Driven Development (TDD)

JUnit

Mocking

## Depuración de errores

---

Se utilizan test para saber si el programa funciona correctamente. Estas pruebas se clasifican en:

- Unitarias: De componentes aislados
- De integración: Varios componentes a la vez
- De validación: Si se cumplen los requisitos
- De sistema: Funcionamiento del sistema completo
- De aceptación: Validación por parte de un usuario
- De regresión: Después de un cambio para verificar que no se han introducido errores
- De carga: Bajo carga normal
- De estrés: Bajo alta carga

## Pruebas de caja

---

### Caja blanca

---

Comprueban el funcionamiento interno y la lógica del código. Tratan de ejecutar todas las sentencias y condiciones al menos una vez construyendo un grafo interno. Esto implica que las pruebas son de complejidad **ciclomática**, ya que el número de caminos posibles en el grafo es enorme

### Caja negra

---

Se guían por los datos de entrada y de salida, sin tener en cuenta los detalles de implementación. Se centran en los requisitos funcionales del software. No se trata de una alternativa a las pruebas de caja blanca, pero son más sencillas. Se basan en comprobar los datos de entrada y salida

## Test Driven Development (TDD)

---

Se basa en el desarrollo de software basado en el siguiente ciclo:

1. Escribir un caso de prueba para una funcionalidad a incluir en el software
2. Implementar el código para que pase el test
3. Refactorizar el código

Sus **ventajas** es la reducción de errores en el código y el desarrollo incremental. Su principal **desventaja** es que el programador de los test también suele implementar el código (error de lógica)

# JUnit

---

Pruebas unitarias sobre métodos aislados

- Si un test falla, los demás no siguen
- `Fail` es utilizado para indicar que una excepción no se ha producido
- `expect` es utilizado para indicar que se va a producir una excepción

## Mocking

---

Pruebas para clases que dependen de otras clases o recursos. Los mock son sustitutos de objetos reales que nos permiten hacer pruebas sin usar el objeto real (puede no estar implementado)