

ESTRUCTURA DE DATOS I

Pruebas unitarias y uso del Depurador



MEng. Andrés Castillo

20/02/24

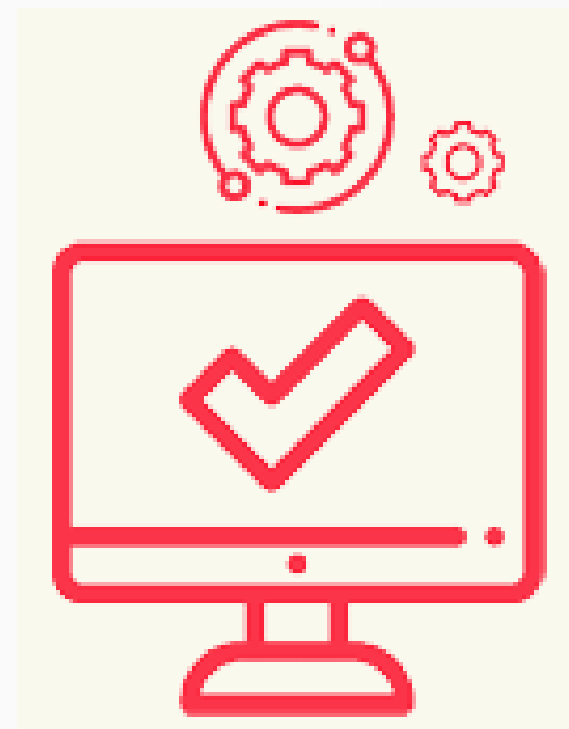
Pruebas de software

- Conceptos que han ganado importancia en el desarrollo de software: **Verificación + Validación = Testing**

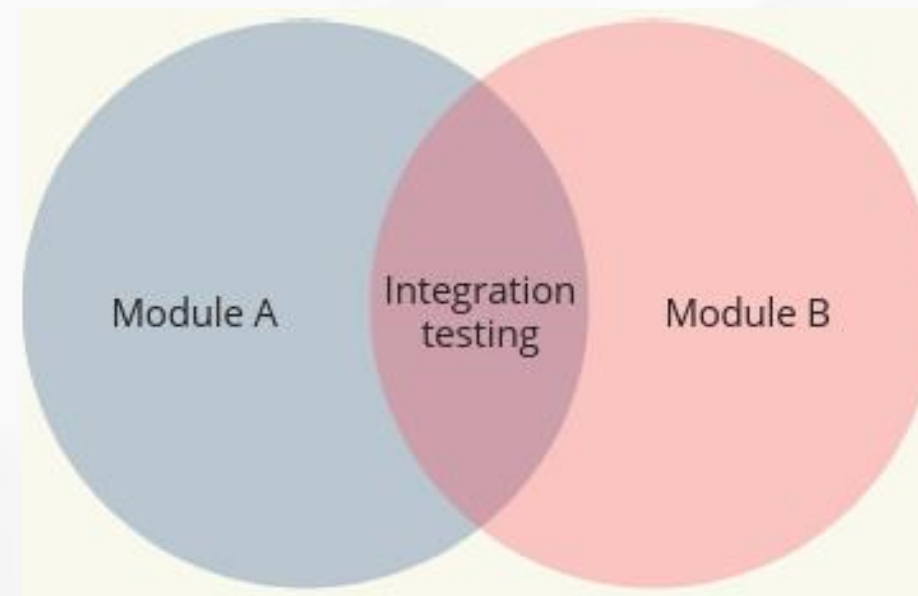
Tipos de pruebas:



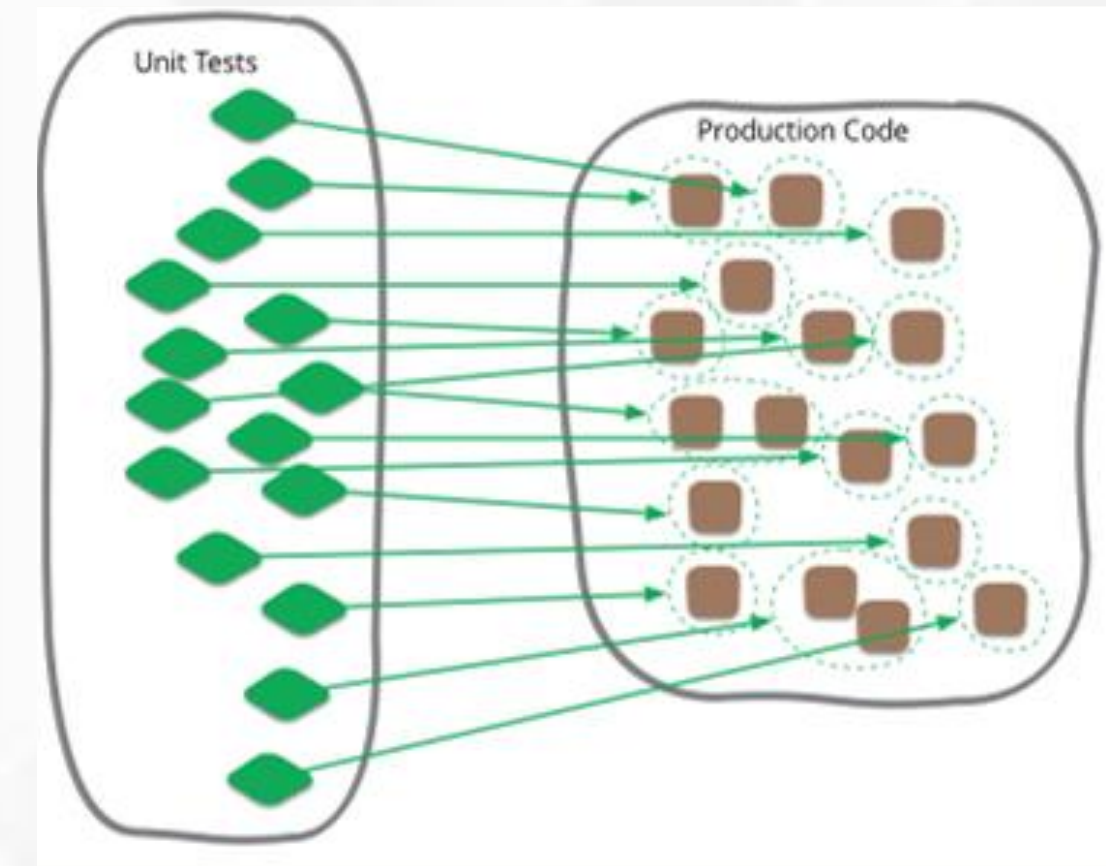
De aceptación



De sistema
"Caja negra"



De integración



Pruebas unitarias
"Caja blanca"

Pruebas unitarias

- Una prueba unitaria es un fragmento de código que verifican un caso concreto de uso de un componente software según las especificaciones.
- Cada prueba se configura para probar un caso determinado de uso de la interfaz de una clase.
- La naturaleza de este tipo de pruebas ha llevado a que sean ampliamente ***automatizadas***.

Beneficios de pruebas unitarias

- Reducir el costo de realización de pruebas.
- Reducir los errores humanos al probar.
- Reducir las diferencias en la calidad de las pruebas que se puede presentar entre diferentes individuos.
- Reducir significativamente el costo de pruebas de regresión.

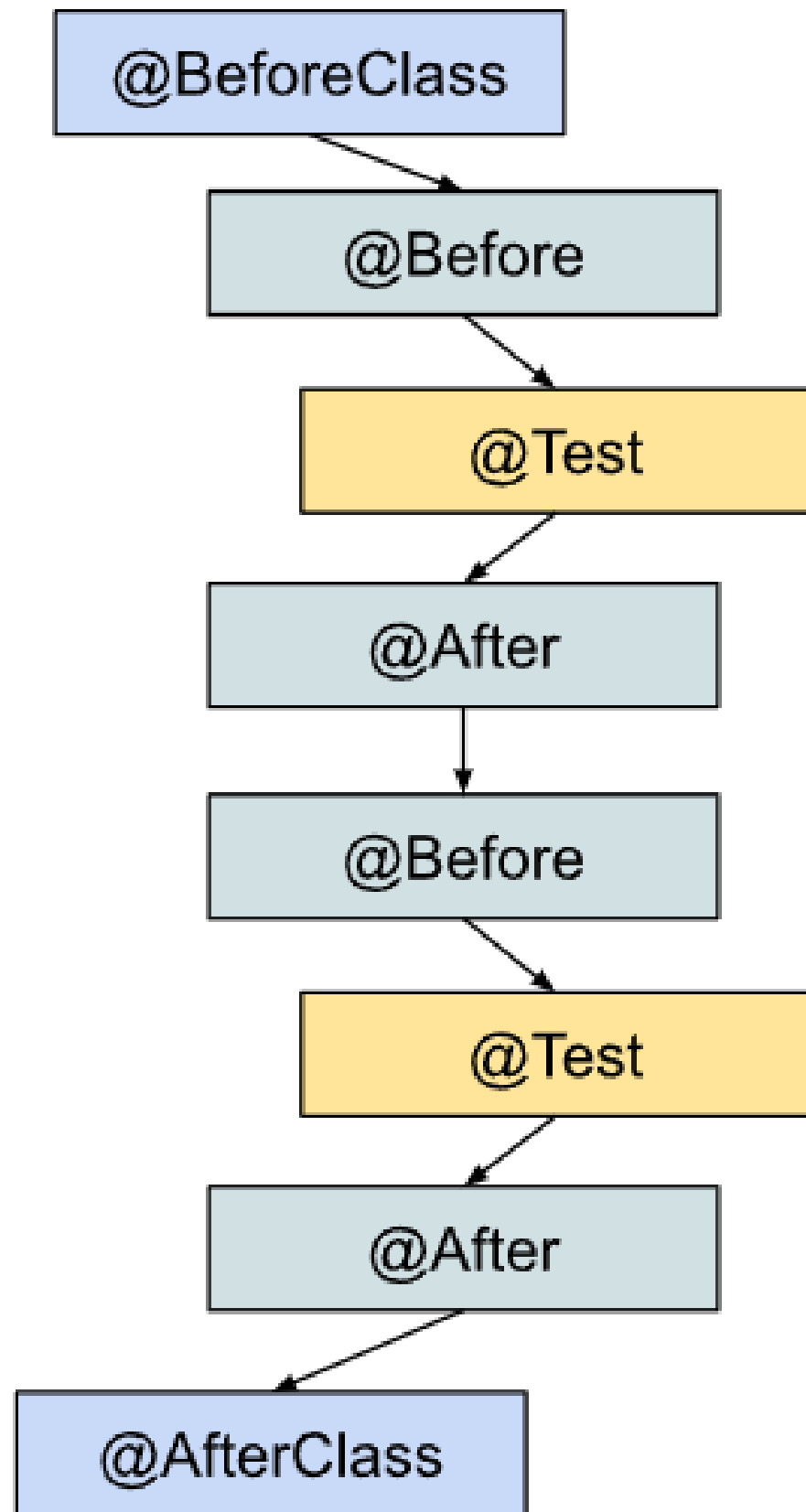
Pruebas unitarias

- Las pruebas se organizan en conjuntos o **suites** de pruebas. Cada suite se asocia a una clase.
- Se prueban, por ejemplo, condiciones o valores límite en argumentos de métodos, o condiciones bajo las que un método genera excepciones.



- En Eclipse se configura en *Project > Properties > Java Build Path > Libraries > Add Library*.

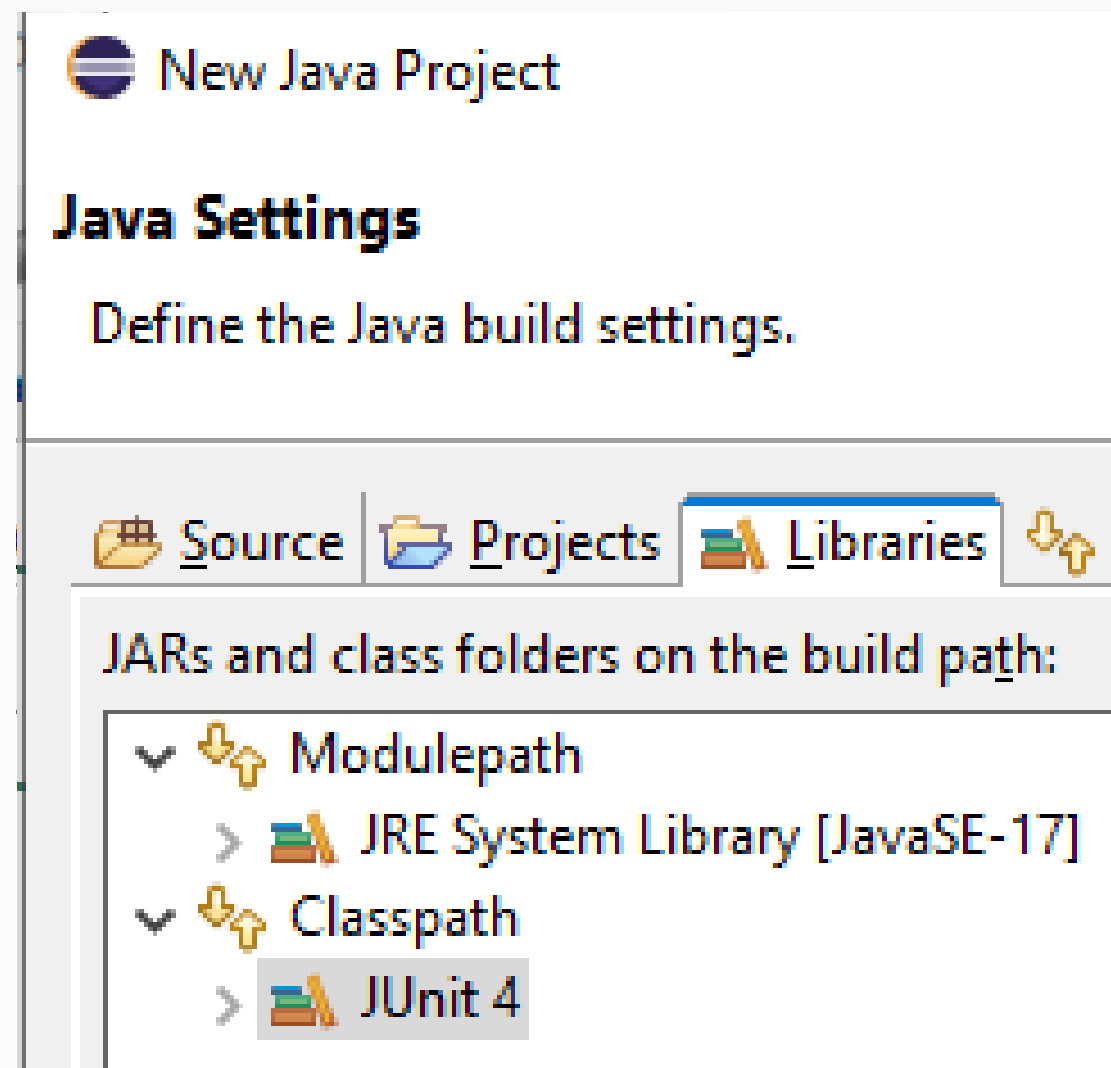
Métodos y anotaciones de JUnit



- ***Cordinate Test.java*** - Abre el test unitario
- Los métodos con anotaciones **@Before** configuran el test. Se ejecutan antes de cada método @Test.
- Los métodos **@Test** contienen una o más pruebas unitarias.
- **assertEquals** comprueba que el valor esperado coincide con el real.
- **assertTrue, assertFalse** comprueban que su argumento devuelve true o false, respectivamente.
- **fail** produce un fallo en el test cuando es ejecutado.

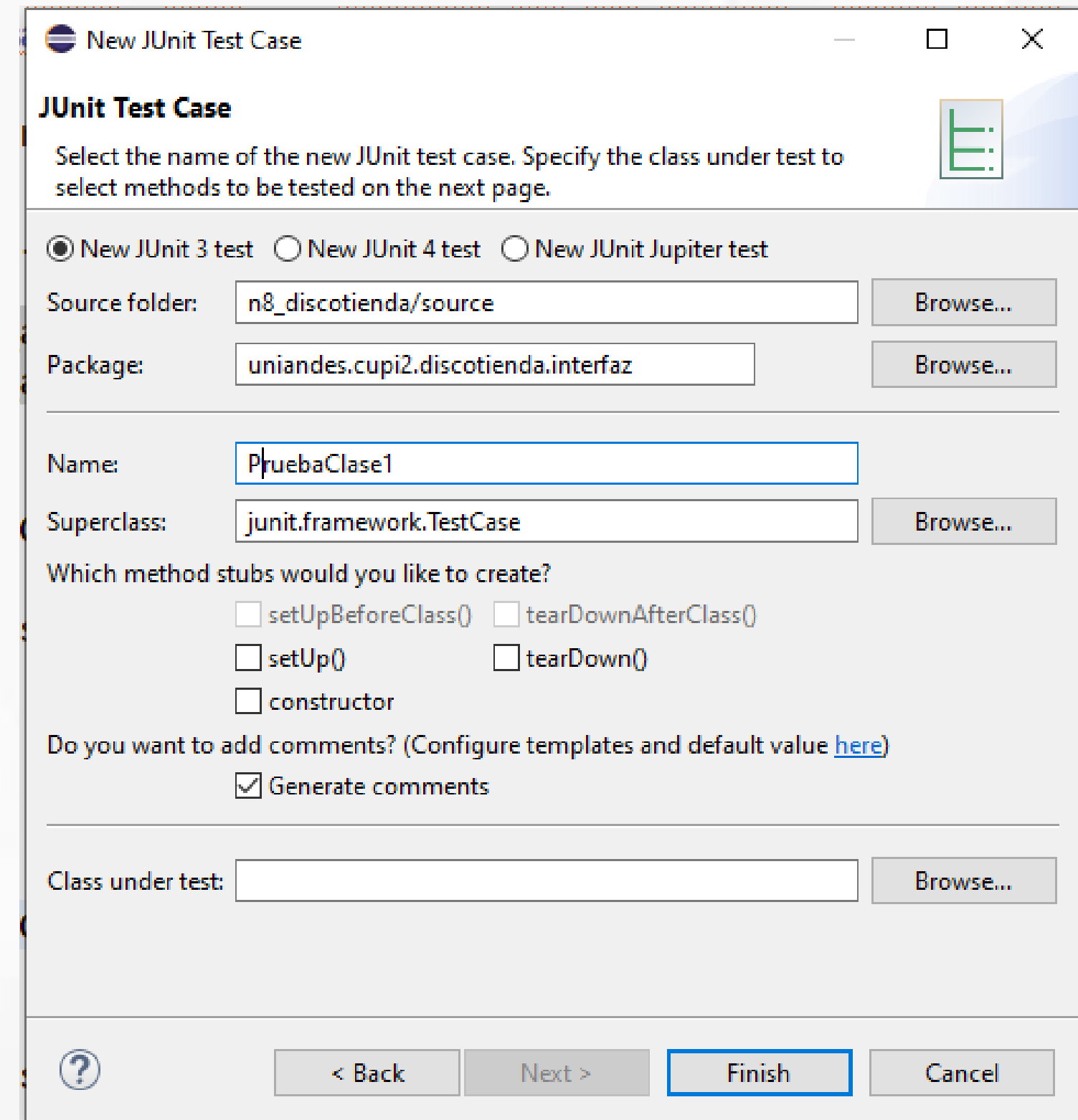
Crear proyecto

1

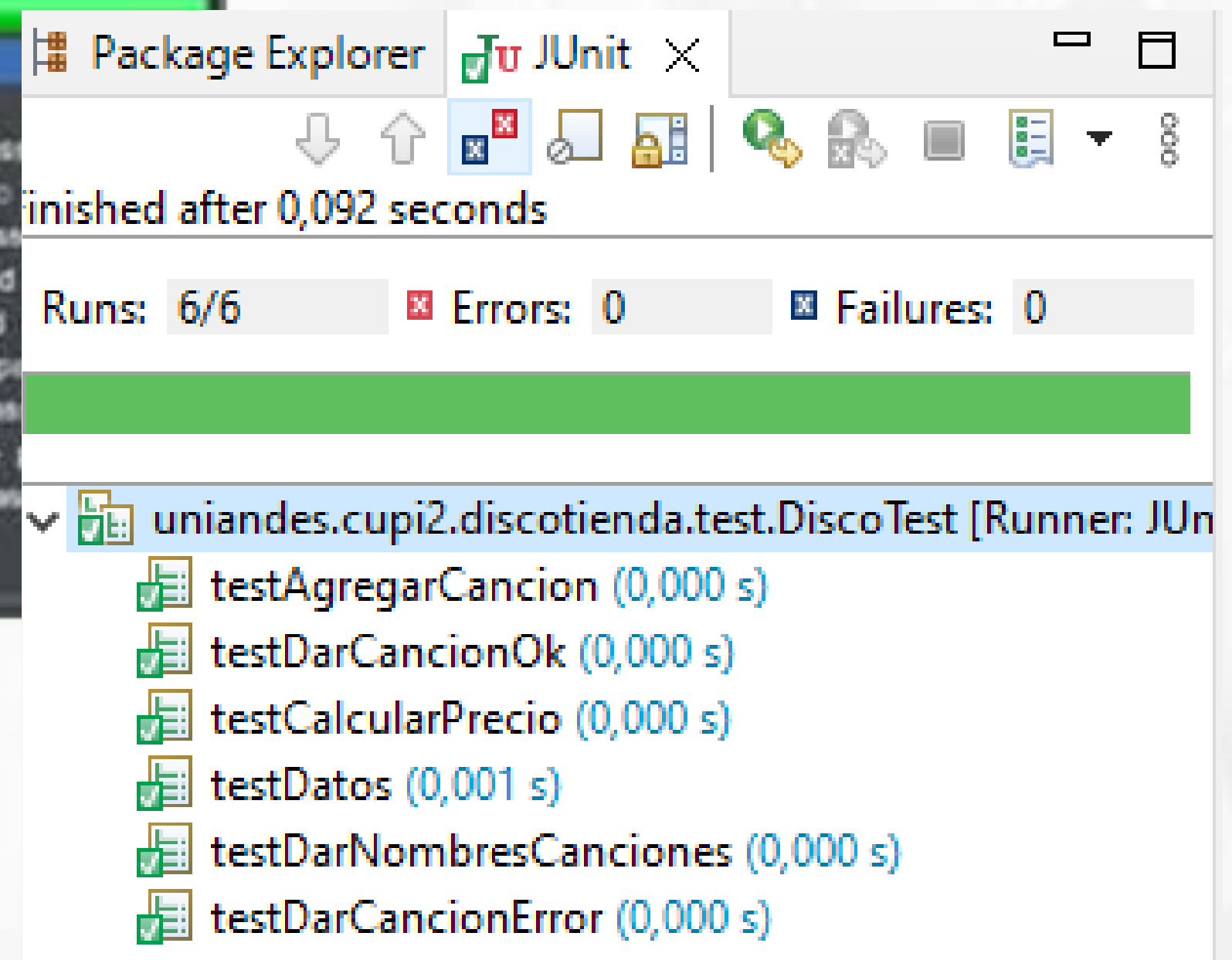
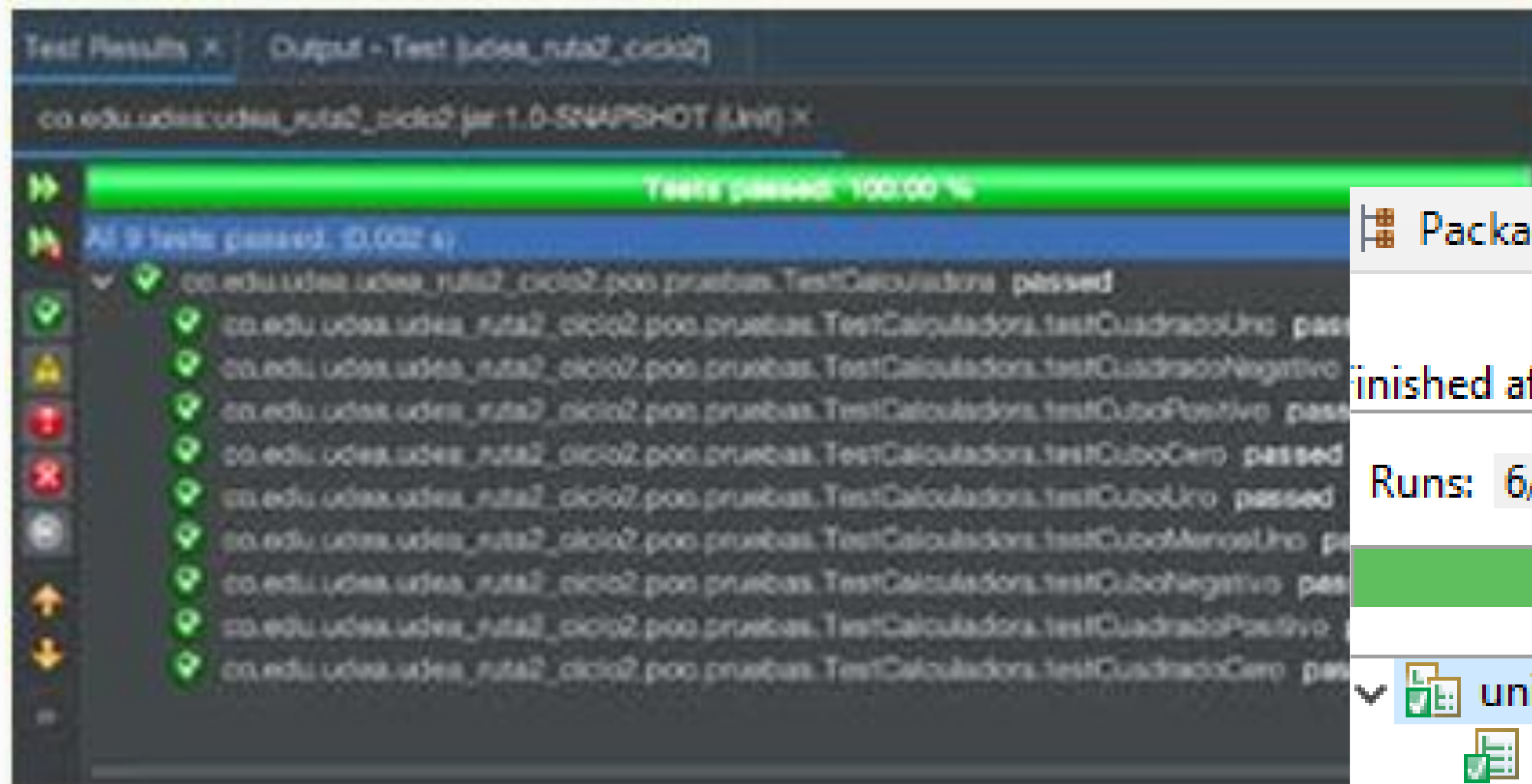


Clase JUnit

2



Run as > JUnit



JUnit 4

<https://junit.org/junit4/>

JUnit 5

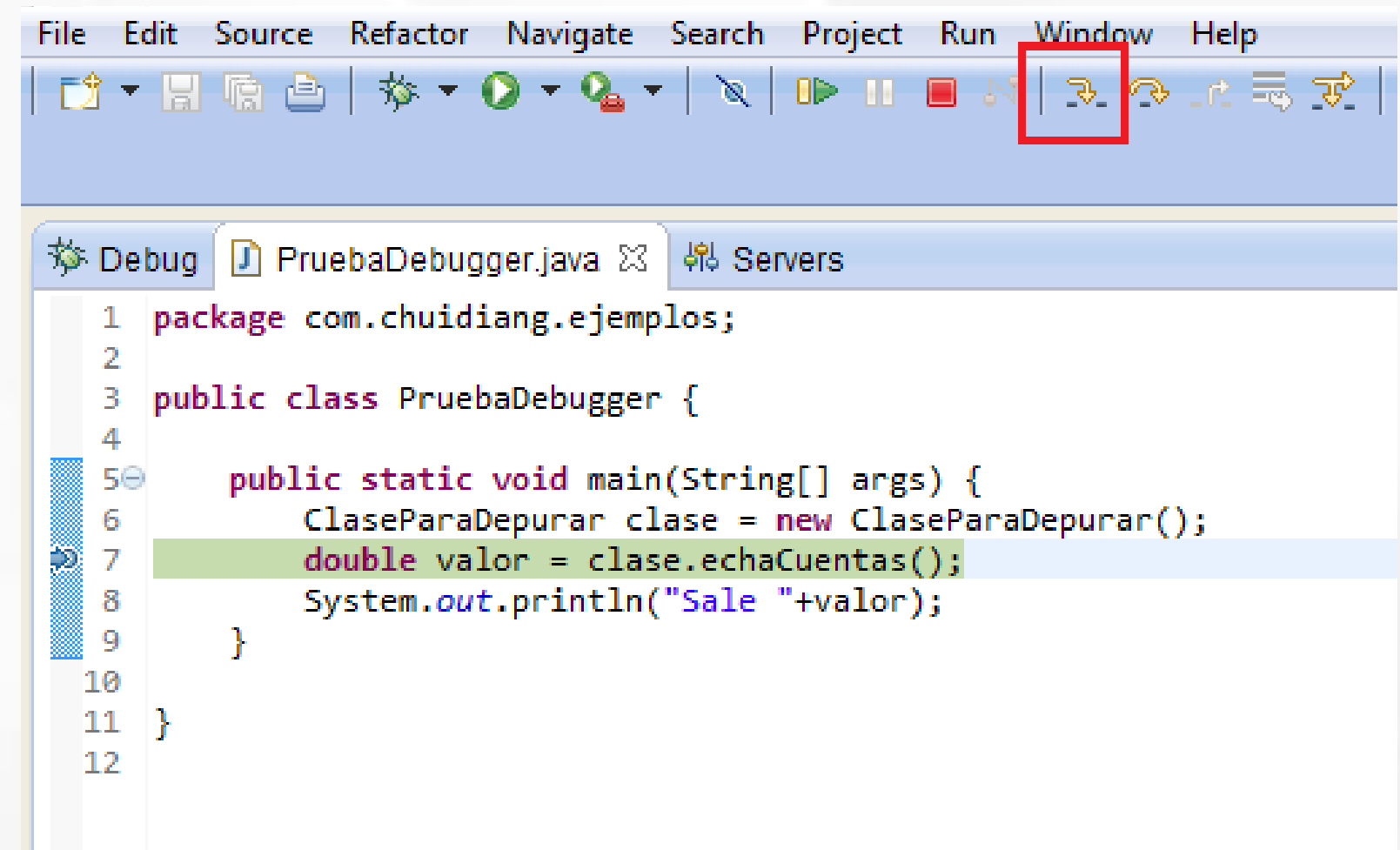
<https://junit.org/junit5/>

Depurador

- En un entorno de desarrollo se refiere al proceso de identificar, analizar y corregir errores o defectos en el código de un programa o aplicación.

Algunas de las actividades comunes:

- Identificación de errores.
- Reproducción del problema.
- Análisis del código.
- Corrección del error.
- Pruebas de regresión.



bugs de debugger



- Dificultan significativamente el proceso de depuración y prolongar el tiempo necesario para identificar y solucionar problemas en el código.
- Los fabricantes de herramientas de desarrollo suelen trabajar para solucionar estos problemas mediante actualizaciones y parches que aborden los bugs conocidos y mejoren la estabilidad y funcionalidad de las herramientas de depuración.

A practicar!!

“La verdad solo se puede encontrar en un lugar: el código”.

Robert C. Martin

Ejemplo



Implementar métodos @Test para los métodos u operaciones de una calculadora.

Reto

Construcción de Pruebas Unitarias - Numeral 5.7
páginas 152-153

Uso básico del depurador - numeral 6
páginas 161-162

Kahoot.it