UD 04.DISEÑO Y REALIZACIÓN DE PRUEBAS

PARTE 2 DE 3: JUNIT EN ECLIPSE

v1.0 19.11.20

ceedcv
CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

Entornos de desarrollo (ED)

Sergio Badal
Carlos Espinosa
c.espinosamoreno@edu.gva.es
Extraído de los apuntes de:
Cristina Álvarez Villanueva; Fco. Javier Valero Garzón; M.ª Carmen Safont

UD 04.DISEÑO Y REALIZACIÓN DE PRUEBAS

- **Pasos a seguir**

1)Lee la documentación (PDF)
2)Instala el software necesario (sigue los pasos)
3)Realiza los TESTS todas las veces que quieras
4)Acude al FORO DE LA UNIDAD
- Para cualquier duda sobre esta unidad

5)Acude al FORO DEL MÓDULO

- Para cualquier duda sobre el módulo

UD 04.DISEÑO Y REALIZACIÓN DE PRUEBAS

- **¿Qué veremos en esta UNIDAD?**

–SEMANA 1

- PARTE 1 DE 3: **CONTROL DE CALIDAD Y PRUEBAS**
- PARTE 2 DE 3: **JUNIT CON ECLIPSE**

–PRÁCTICA NO **EVALUABLE**

–SEMANA 2

- PARTE 3 DE 3: **JUNIT CON NETBEANS**

–PRÁCTICA NO **EVALUABLE**

4.2 JUNIT EN ECLIPSE

JUNIT EN ECLIPSE

- En prácticas anteriores hemos hecho **pruebas unitarias de caja blanca de forma MANUAL** a partir de un código mediante los depuradores de NetBeans y Eclipse.

- En esta unidad vamos a hacer **pruebas unitarias de caja negra de forma AUTOMÁTICA** con la herramienta JUnit.

  - Recuerda que las pruebas unitarias se hacían sobre una clase para observar su comportamiento independientemente del resto de clases de la aplicación.

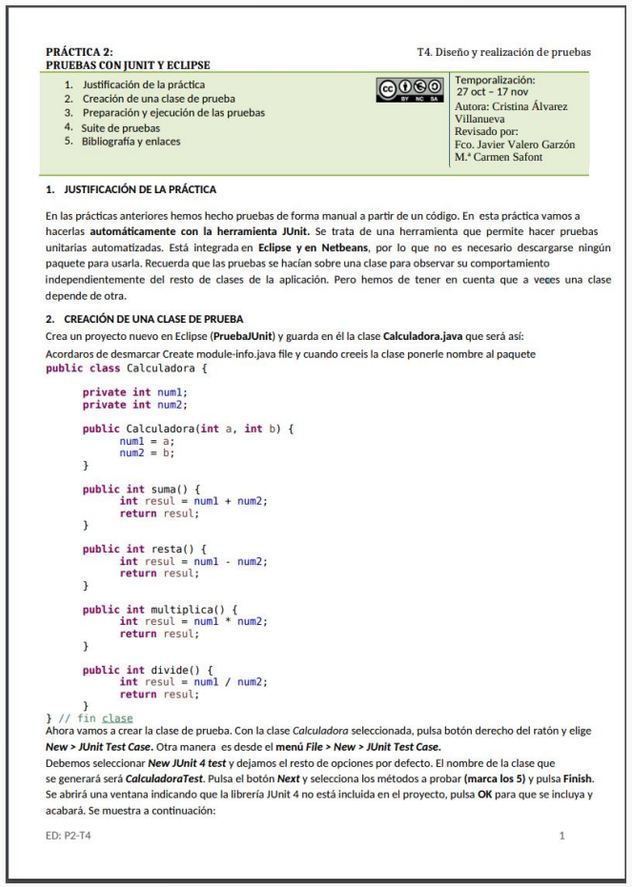
  - Está integrada en Eclipse y en Netbeans, por lo que no es necesario descargarse ningún paquete para usarla.

- **Veremos esta semana JUNIT en Eclipse y la que viene en NetBeans.**

4.2 JUNIT EN ECLIPSE

JUNIT EN ECLIPSE

- Para usar JUNIT en Eclipse te recomendamos que sigas este tutorial creado por los profesores del curso pasado para probar una **clase de una calculadora sencilla**.
  - Disponible en el Aula Virtual
- Complementariamente, puedes ver este vídeo creado para este curso:
  - Disponible en el Aula Virtual
- Adicionalmente, puedes ver estos otros vídeos externos:
  - JUnit en Eclipse básico: https://www.youtube.com/watch?v=1k22KuD4si0
  - JUnit en Eclipse avanzado: https://www.youtube.com/watch?v=ql0BX4hq6D4
- Como práctica no evaluable tendrás que realizar algo similar para comprobar que has entendido cómo funcionan este tipo de pruebas unitarias. Lo vemos en la siguiente página.



PRÁCTICA 2:
PRUEBAS CON JUNIT Y ECLIPSE

T4. Diseño y realización de pruebas

1. Justificación de la práctica
2. Creación de una clase de prueba
3. Preparación y ejecución de las pruebas
4. Suite de pruebas
5. Bibliografía y enlaces

Temporalización:
27 oct – 17 nov

Autora: Cristina Álvarez Villanueva
Revisado por:
Fco. Javier Valero Garzón
M.ª Carmen Safont

1. JUSTIFICACIÓN DE LA PRÁCTICA

En las prácticas anteriores hemos hecho pruebas de forma manual a partir de un código. En esta práctica vamos a hacerlas **automáticamente con la herramienta JUnit.** Se trata de una herramienta que permite hacer pruebas unitarias automatizadas. Está integrada en **Eclipse y en Netbeans**, por lo que no es necesario descargarse ningún paquete para usarla. Recuerda que las pruebas se hacían sobre una clase para observar su comportamiento independientemente del resto de clases de la aplicación. Pero hemos de tener en cuenta que a veces una clase depende de otra.

2. CREACIÓN DE UNA CLASE DE PRUEBA

Crea un proyecto nuevo en Eclipse (**PruebaJUnit**) y guarda en él la clase **Calculadora.java** que será así:

Acordaros de desmarcar Create module-info.java file y cuando creeis la clase ponerle nombre al paquete

```
public class Calculadora {

	private int num1;
	private int num2;

	public Calculadora(int a, int b) {
		num1 = a;
		num2 = b;
	}

	public int suma() {
		int resul = num1 + num2;
		return resul;
	}

	public int resta() {
		int resul = num1 - num2;
		return resul;
	}

	public int multiplica() {
		int resul = num1 * num2;
		return resul;
	}

	public int divide() {
		int resul = num1 / num2;
		return resul;
	}
} // fin clase
```

Ahora vamos a crear la clase de prueba. Con la clase *Calculadora* seleccionada, pulsa botón derecho del ratón y elige ***New > JUnit Test Case.*** Otra manera es desde el **menú *File > New > JUnit Test Case.***

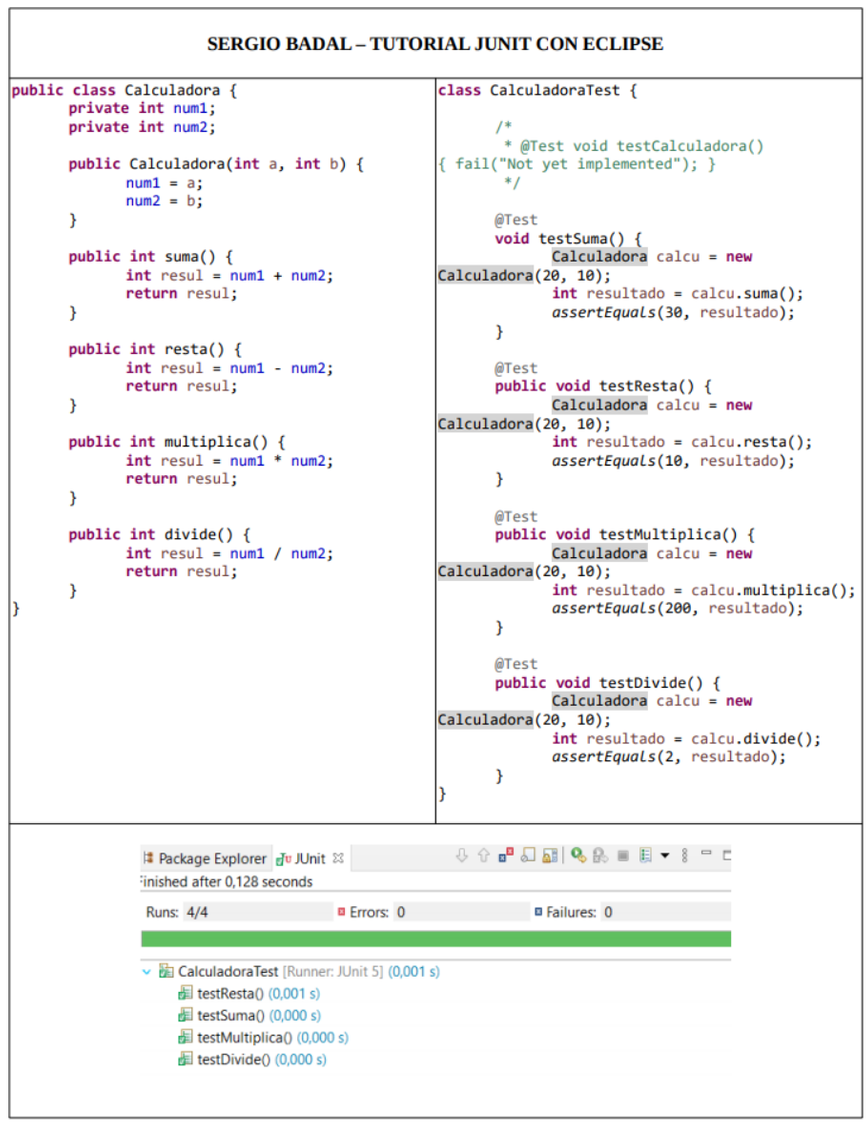
Debemos seleccionar ***New JUnit 4 test*** y dejamos el resto de opciones por defecto. El nombre de la clase que se generará será ***CalculadoraTest***. Pulsa el botón ***Next*** y selecciona los métodos a probar **(marca los 5)** y pulsa **Finish**. Se abrirá una ventana indicando que la librería JUnit 4 no está incluida en el proyecto, pulsa **OK** para que se incluya y acabará. Se muestra a continuación:

ED: P2-T4 1

4.2 JUNIT EN ECLIPSE

PRÁCTICA NO EVALUABLE

- Para validar que has entendido el proceso, te recomendamos que repitas el proceso creando una clase **micalculadora** con estos métodos:
  - Constructor que recibe TRES datos enteros
  - Método **cartesiano** que devuelve el producto de los tres datos (entero)
  - Método **sumatriple** que devuelve la suma de los tres datos (entero)
  - Método **media** que devuelve la media de los tres (float)
  - Método **mayordelostres** que devuelve el mayor de los tres (int)
- **Si no sabes programar en Java busca en Google o acude al foro y te ayudamos :-)**
- Una vez lo tengas, crea los métodos JUnit y entrega un PDF **como el de la derecha** con estos datos:
  - Código de la clase
  - Código de la clase JUnit
  - Captura del resultado



SERGIO BADAL – TUTORIAL JUNIT CON ECLIPSE

```
public class Calculadora {
	private int num1;
	private int num2;

	public Calculadora(int a, int b) {
		num1 = a;
		num2 = b;
	}

	public int suma() {
		int resul = num1 + num2;
		return resul;
	}

	public int resta() {
		int resul = num1 - num2;
		return resul;
	}

	public int multiplica() {
		int resul = num1 * num2;
		return resul;
	}

	public int divide() {
		int resul = num1 / num2;
		return resul;
	}
}
```

```
class CalculadoraTest {

	/*
	 * @Test void testCalculadora()
{ fail("Not yet implemented"); }
	 */

	@Test
	void testSuma() {
		Calculadora calcu = new
Calculadora(20, 10);
		int resultado = calcu.suma();
		assertEquals(30, resultado);
	}

	@Test
	public void testResta() {
		Calculadora calcu = new
Calculadora(20, 10);
		int resultado = calcu.resta();
		assertEquals(10, resultado);
	}

	@Test
	public void testMultiplica() {
		Calculadora calcu = new
Calculadora(20, 10);
		int resultado = calcu.multiplica();
		assertEquals(200, resultado);
	}

	@Test
	public void testDivide() {
		Calculadora calcu = new
Calculadora(20, 10);
		int resultado = calcu.divide();
		assertEquals(2, resultado);
	}
}
```