

INTRODUCCIÓN A UNIT TESTING

- Unit Testing es un tipo de prueba que valida componentes individuales del software.
- Cada unidad (método, clase) se prueba de forma aislada para asegurar su correcto funcionamiento.
- Es la base para garantizar calidad y reducir errores en etapas tempranas.

OBJETIVOS DEL UNIT TESTING

- Verificar que cada unidad del código funciona como se espera.
- Facilitar la detección temprana de errores y reducir costes de corrección.
- Permitir refactorización segura y desarrollo ágil.

VENTAJAS

- Aumenta la calidad del software y la confianza en el código.
- Reduce el tiempo y coste de mantenimiento.
- Sirve como documentación viva del comportamiento esperado.

DESVENTAJAS

- Requiere tiempo y esfuerzo inicial para escribir pruebas.
- Puede ser complejo en sistemas con alta dependencia entre módulos.
- No sustituye pruebas de integración o aceptación.

HERRAMIENTAS Y EJEMPLO EN JAVA

- JUnit es el framework estándar para pruebas unitarias en Java.
- Permite automatizar pruebas y ejecutarlas en entornos CI/CD (**Continuous Integration / Continuous Delivery (o Deployment)**).
- Ejemplo: `@Test` para validar métodos individuales.

A blue callout box contains the following text:

No lo hemos visto en clase...
permite importar elementos estáticos de una clase
para nombrarlos directamente
[-- assertEquals() en vez de Assert.assertEquals() --]

The code snippet is enclosed in a light blue box:

```
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculadoraTest {

    @Test
    public void testSuma() {
        int resultado = Calculadora.sumar(2, 3);
        assertEquals(5, resultado);
    }
}
```