



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ Pruebas Unitarias en Java

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Casos de Prueba con Junit

LEARNING PATHWAY

4.7

Start! 🚩

Pruebas Unitarias en Java

Integración JUnit con
Eclipse

Cobertura

Integración JUnit con Eclipse

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Aprender a integrar JUnit con Eclipse IDE





Rompehielo 🧊



Respondan en el chat o levantando la mano 🙋

1. ¿Dónde crees que se está haciendo click para desplegar el menú de la imagen?
2. ¿Piensas que será necesario instalar algún plugin? ¿Por qué?



› Integración de JUnit con Eclipse

Integración de JUnit con Eclipse

Aprovechando las ventajas del IDE:

Una de las principales ventajas de desarrollar java en Eclipse, es que es un IDE con múltiples integraciones, una de ellas es JUnit.

En las últimas versiones de Eclipse y en la mayoría de los casos, no es necesario ningún tipo de integración adicional. Solo deberemos tener en cuenta algunas consideraciones como las que veremos a continuación.

A screenshot of the Eclipse IDE interface. The top of the window shows two tabs: 'JUnitTesting.java' and 'squareTest.java'. The 'squareTest.java' tab is active, displaying the following Java code:

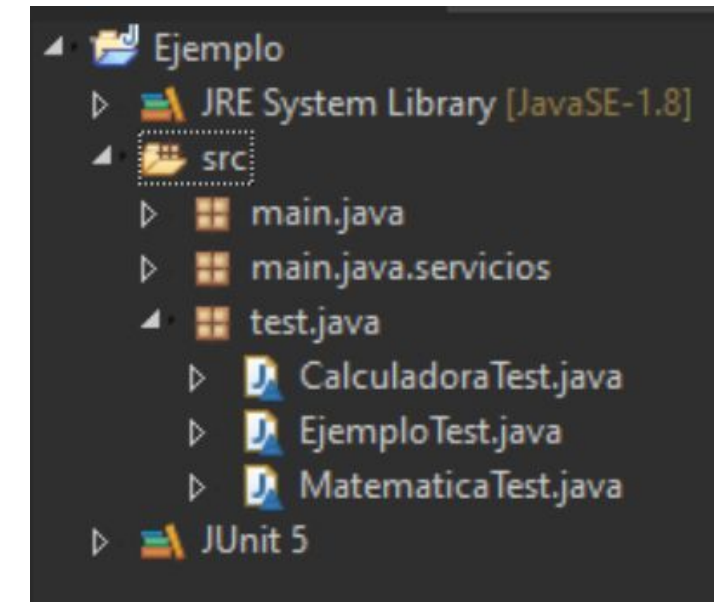
```
1 package testing;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class squareTest {
8
9     @Test
10     public void test() {
11         fail("Not yet implemented");
12     }
13
14 }
15
```



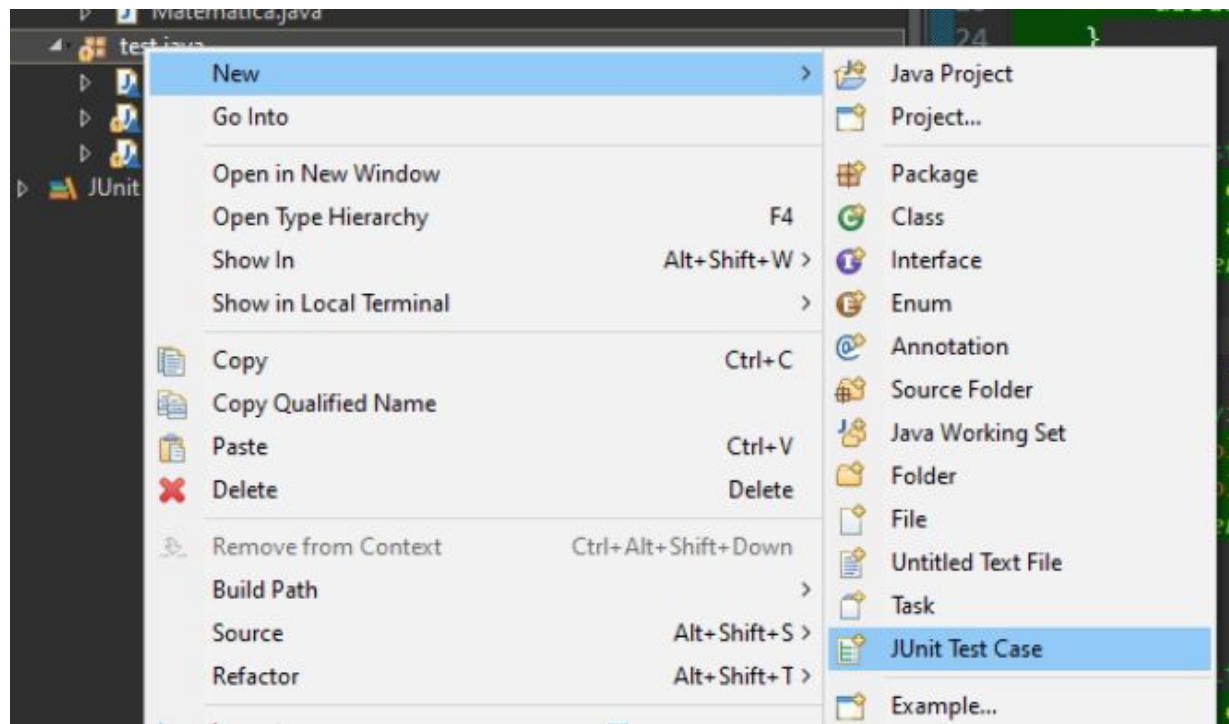
Integración de JUnit con Eclipse

A tener en cuenta:

- **Versión de JUnit:** Es recomendable utilizar JUnit 5, pero si se implementa una versión anterior, deberás prestar atención a las @Anotaciones a usar, ya que pueden variar.
- **Gestión de Paquetes:** Es necesario que la estructura del proyecto esté ordenada. Para esto es importante tener los Test en un paquete diferenciado del resto de la app, como podemos ver en la imagen: dentro de main tendremos nuestra App en sí, con su organización de paquetes determinada. Por otro lado el **paquete test** tendrá un sub paquete llamado java, que es donde se contendrán los clases Test.



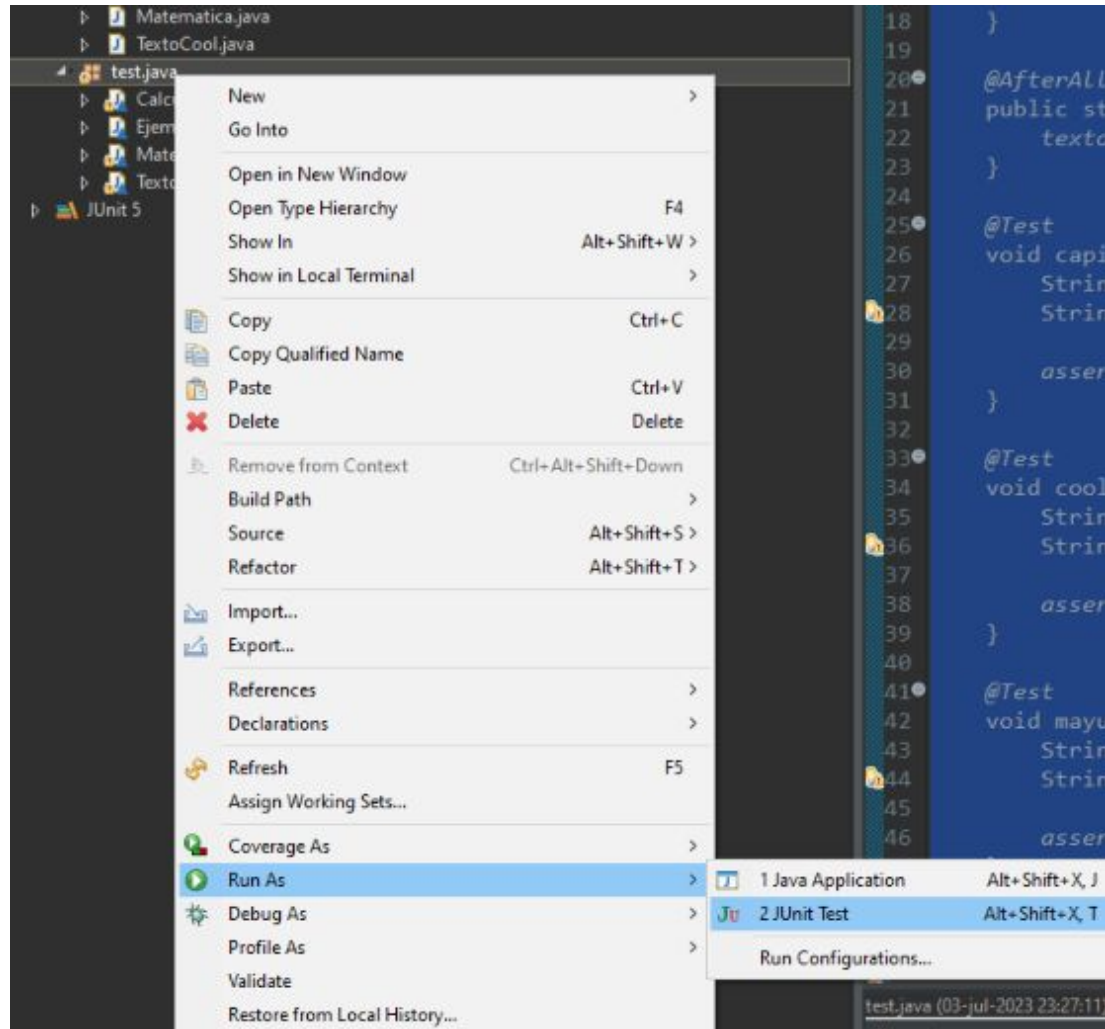
Integración de JUnit con Eclipse



A tener en cuenta:

- **Generate Test Cases:** La manera más sencilla de generar un test case o ClaseTest, es haciendo click derecho sobre el paquete donde quiero generarlo y colocarle el nombre correspondiente **[NombreOriginalDeLaClase]+Test.**

Integración de JUnit con Eclipse



A tener en cuenta:

- **Corriendo los Test:** Para correr todos los test desde Eclipse, puede hacer click derecho sobre el paquete de Test y seleccionar Run As > JUnit Test.

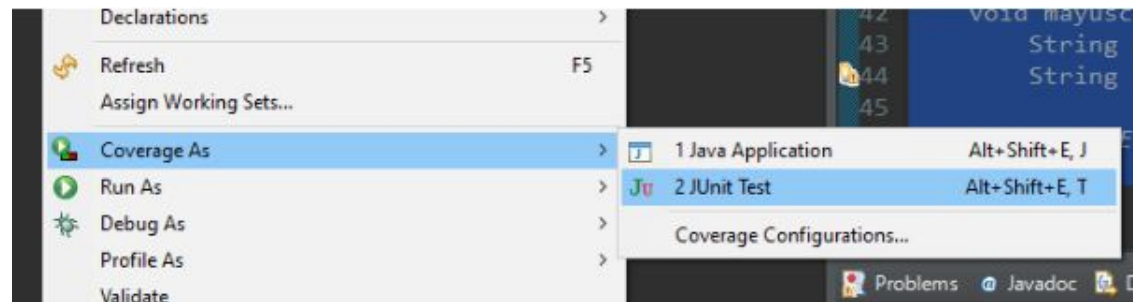
En caso de querer correr los test de una sola clase, realizar el click derecho desde la clase deseada.



Integración de JUnit con Eclipse



- **Corriendo los Test:** Si se desea ver el coverage de los test realizados, debes hacer click derecho y seleccionar “**Coverage As**”. Se desplegará la siguiente pantalla.



test.java (03-jul-2023 23:43:19)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▲ Ejemplo	<div><div></div></div> 90,6 %	329	34	363
▲ src	<div><div></div></div> 90,6 %	329	34	363
▲ main.java.servicios	<div><div></div></div> 83,8 %	155	30	185
▶ Calculadora.java	<div><div></div></div> 0,0 %	0	30	30
▶ Matematica.java	<div><div></div></div> 100,0 %	36	0	36
▶ TextoCool.java	<div><div></div></div> 100,0 %	119	0	119
▲ main.java	<div><div></div></div> 0,0 %	0	4	4
▶ Ejemplo.java	<div><div></div></div> 0,0 %	0	4	4
▲ test.java	<div><div></div></div> 100,0 %	174	0	174
▶ CalculadoraTest.java	<div><div></div></div> 100,0 %	4	0	4
▶ EjemploTest.java	<div><div></div></div> 100,0 %	25	0	25
▶ MatematicaTest.java	<div><div></div></div> 100,0 %	104	0	104
▶ TextoCoolTest.java	<div><div></div></div> 100,0 %	41	0	41

Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE CODING

Ejemplo en vivo

¡Vamos a comenzar la estructura para testear!:

En este ejemplo vamos a crear los paquetes y clases de Test necesarios para algunos ejercicios anteriores que hemos realizado.

1. *Crear un método a evaluar en cada clase Test creada y mostrar cómo correrlo en Eclipse **con** y **sin** porcentaje de cobertura.*

 **Tiempo: 25 minutos**





Ejercicio N° 1

Cobertura



Cobertura

Es hora de cubrir nuestros proyectos: 🙌

Para poner en práctica lo aprendido, deberás crear los paquetes y clases Test necesarios para los ejercicios realizados en el módulo anterior (Polimorfismo).

Consigna: ✍️

Deberás crear la mayor cantidad de clases Test posibles en el tiempo estimado, para tener ya lista la estructura necesaria así más adelante podrás comenzar a testear los métodos implementados.



Tiempo 🕒: 25 minutos

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender cómo implementar JUnit en Eclipse IDE**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌🙌🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. *Lectura Módulo 4, Lección 7: páginas 8 - 10*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

