



El futuro digital
es de todos

MinTIC

JUnit

Documentación y Configuración
JUNIT - Parte 1



Universidad de Caldas

Hola:

En la anterior ocasión se analizó la calidad de *software*, un concepto determinante a la hora de categorizar a un desarrollador, que en algunas ocasiones deja la calidad a un lado y prioriza el funcionamiento del *software* con pruebas manuales y básicas. Por el contrario, la gran mayoría de desarrolladores enfocan la calidad del *software* como pilar fundamental en el funcionamiento de una aplicación que es desarrollada individualmente o en equipo a través de una metodología llamada **Desarrollo Dirigido por Pruebas**, donde lo primero que se hace son las pruebas y, posteriormente, se implementa el resto del sistema, que es algo similar a lo que se ha mostrado en la parte práctica.

Las **pruebas unitarias** son un tipo de pruebas de *software* específico que se ejecutan sobre porciones pequeñas de código con funcionalidades muy detalladas. A continuación está la documentación asociada a la librería JUNIT, que brindará soporte en este proceso.

Video de documentación en el sitio web de JUNIT y todos sus componentes.

The screenshot shows the JUnit 5 website (<https://junit.org/junit5/>) displayed in a web browser. The page features a large header with the text "JUnit 5" and "The 5th major version of the programmer-friendly testing framework for Java and the JVM". Below the header are several navigation links: "User Guide", "Javadoc", "Code & Issues", "Q & A", and "Support JUnit". The main content area includes sections for "About", "Resources", "Sponsoring", "Latest Release", "Upcoming Events", and "Thank You". The "About" section mentions the goal of creating an up-to-date foundation for developer-side testing on the JVM. The "Resources" section provides links to the User Guide, Javadoc, and GitHub Repository. The "Sponsoring" section asks users to support the project. The "Latest Release" section highlights the Jupiter v5.7.2, Vintage v5.7.2, and Platform v1.7.2 releases. The "Upcoming Events" section notes there are no events at the moment. The "Thank You" section thanks sponsors like GitHub and Gradle Enterprise.

The screenshot shows the JUnit 5.7.2 API documentation (<https://junit.org/junit5/docs/current/api/>) displayed in a web browser. The page has a green header bar with tabs for "OVERVIEW", "MODULE", "PACKAGE", "CLASS", "USE", "TREE", "DEPRECATED", "INDEX", and "HELP". The main content area displays a table of JUnit modules with their descriptions:

Module	Description
<code>org.junit.jupiter</code>	Aggregates all JUnit Jupiter modules.
<code>org.junit.jupiter.api</code>	Defines JUnit Jupiter API for writing tests.
<code>org.junit.jupiter.engine</code>	Provides the JUnit Jupiter <code>TestEngine</code> implementation.
<code>org.junit.jupiter.migrationsupport</code>	Support for migrating from JUnit 4 to JUnit Jupiter.
<code>org.junit.jupiter.params</code>	JUnit Jupiter extension for parameterized tests.
<code>org.junit.platform.commons</code>	Common APIs and support utilities for the JUnit Platform.
<code>org.junit.platform.console</code>	Support for launching the JUnit Platform from the console.
<code>org.junit.platform.engine</code>	Public API for test engines.
<code>org.junit.platform.jfr</code>	Provides the JUnit Platform Flight Recording Listener.
<code>org.junit.platform.launcher</code>	Public API for configuring and launching test plans.
<code>org.junit.platform.reporting</code>	Defines the JUnit Platform Reporting API.
<code>org.junit.platform.runner</code>	Runner and annotations for configuring and executing tests on the JUnit Platform in a JUnit 4 environment.
<code>org.junit.platform.suite.api</code>	Annotations for configuring a test suite on the JUnit Platform.
<code>org.junit.platform.testkit</code>	Defines the Test Kit API for the JUnit Platform.

The screenshot shows the GitHub repository page for `junit-team/junit5`. The page includes a navigation bar with links to Why GitHub?, Team, Enterprise, Explore, Marketplace, Pricing, a search bar, and sign-in options. Below the header, there's a repository summary with a link to the code, 119 issues, 11 pull requests, discussions, actions, wiki, security, and insights. A sidebar on the left shows the main branch dropdown set to 'main', 42 branches, and 53 tags. The main content area displays a list of recent commits from juliette-derancourt, including support for AutoCloseable, GH Actions workflow, and updates to ArchUnit and Gradle. To the right, there's an 'About' section with a checked checkbox stating 'The 5th major version of the programmer-friendly testing framework for Java and the JVM', a link to `junit.org`, and a list of tags: kotlin, java, junit, test-framework, kotlin-testing, junit-vintage, junit-jupiter, and junit-platform.

The screenshot shows the `README.md` file for the `junit5` repository. It features a title 'Continuous Integration Builds' with two green status indicators: 'CI passing' and 'Cross-Version passing'. Below the title, there's a paragraph about the official CI build server for JUnit 5, mentioning it performs quick checks on pull requests and build matrices across various OpenJDK versions. The 'Code Coverage' section explains that JaCoCo is used for coverage, with a link to Codecov. It also provides instructions for generating a local coverage report using the Gradle Wrapper. The 'Gradle Build Scans and Build Caching' section is present at the bottom. A footer note at the very bottom states 'Revved up by Gradle Enterprise'.

The screenshot shows the JUnit 5 User Guide homepage. The left sidebar contains a 'Table of Contents' with sections like Overview, Writing Tests, and API Evolution. The main content area features a large title 'JUnit 5 User Guide' and a sub-section '1. Overview'. It includes a brief introduction, a note about PDF download, and detailed descriptions of the JUnit Platform, Jupiter, and Vintage sub-projects. A 'Reproductor de video' button is visible on the right.

JUnit 5 User Guide

Stefan Bechtold · Sam Brannen · Johannes Link · Matthias Merdes · Marc Philipp · Juliette de Rancourt · Christian Stein
– Version 5.7.2

1. Overview

The goal of this document is to provide comprehensive reference documentation for programmers writing tests, extension authors, and engine authors as well as build tool and IDE vendors.

This document is also available as a [PDF download](#).

1.1. What is JUnit 5?

Unlike previous versions of JUnit, JUnit 5 is composed of several different modules from three different sub-projects.

JUnit 5 = JUnit Platform + JUnit Jupiter + JUnit Vintage

The **JUnit Platform** serves as a foundation for launching testing frameworks on the JVM. It also defines the [TestEngine](#) API for developing a testing framework that runs on the platform. Furthermore, the platform provides a [Console Launcher](#) to launch the platform from the command line and a [JUnit 4 based Runner](#) for running any [TestEngine](#) on the platform in a JUnit 4 based environment. First-class support for the JUnit Platform also exists in popular IDEs (see [IntelliJ IDEA](#), [Eclipse](#), [NetBeans](#), and [Visual Studio Code](#)) and build tools (see [Gradle](#), [Maven](#), and [Ant](#)).

JUnit Jupiter is the combination of the new [programming model](#) and [extension model](#) for writing tests and extensions in JUnit 5. The Jupiter sub-project provides a [TestEngine](#) for running Jupiter based tests on the platform.

JUnit Vintage provides a [TestEngine](#) for running JUnit 3 and JUnit 4 based tests on the platform.

[Reproductor de video](#)

The screenshot shows a specific page from the JUnit 5 User Guide titled '1.4.2. JUnit 5 Features'. The left sidebar is identical to the homepage. The main content area starts with a section on available features, followed by a list of links to various resources like 'Writing Tests in JUnit Jupiter' and 'Extension Model for JUnit Jupiter'. Below this is a section on 'Example Projects' with a list of projects for different build tools and languages. The page concludes with a section on '2. Writing Tests' and a note about minimum requirements.

1.4.2. JUnit 5 Features

To find out what features are available in JUnit 5 and how to use them, read the corresponding sections of this User Guide, organized by topic.

- [Writing Tests in JUnit Jupiter](#)
- [Migrating from JUnit 4 to JUnit Jupiter](#)
- [Running Tests](#)
- [Extension Model for JUnit Jupiter](#)
- Advanced Topics
 - [JUnit Platform Launcher API](#)
 - [JUnit Platform Test Kit](#)

1.4.3. Example Projects

To see complete, working examples of projects that you can copy and experiment with, the `junit5-samples` repository is a good place to start. The `junit5-samples` repository hosts a collection of sample projects based on JUnit Jupiter, JUnit Vintage, and other testing frameworks. You'll find appropriate build scripts (e.g., `build.gradle`, `pom.xml`, etc.) in the example projects. The links below highlight some of the combinations you can choose from.

- For Gradle and Java, check out the [junit5-jupiter-starter-gradle](#) project.
- For Gradle and Kotlin, check out the [junit5-jupiter-starter-gradle-kotlin](#) project.
- For Gradle and Groovy, check out the [junit5-jupiter-starter-gradle-groovy](#) project.
- For Maven, check out the [junit5-jupiter-starter-maven](#) project.
- For Ant, check out the [junit5-jupiter-starter-ant](#) project.

2. Writing Tests

The following example provides a glimpse at the minimum requirements for writing a test in JUnit Jupiter. Subsequent sections of this chapter will provide further details on all available features.



**Mision
TIC 2022**

The logo features the text "Mision TIC 2022" in a bold, sans-serif font. The word "Mision" is in blue, "TIC" is in red, and "2022" is in blue. A red curved line starts from the top of the letter "i" in "Mision" and ends at the top of the letter "i" in "2022". The background of the logo is a white circle with a gray halftone pattern, set against a dark red circular frame.

Universidad de Caldas