



PROYECTO JUNIT

Entornos de Desarrollo

DESCRIPCIÓN BREVE

Este proyecto desarrolla una librería en Java con seis métodos distintos, cada uno abordando una categoría específica. Se incluyen pruebas unitarias con JUnit 5 y se utiliza Gradle en IntelliJ para gestionar el proyecto y generar documentación detallada en formato HTML con JavaDocs.

Andreu Orenga Ramon
1ºDAW

INDICE

1	Objetivo	2
2	Categorización de Métodos:.....	2
2.1	Métodos Numéricos: factorial(int n)	2
2.2	Métodos de Texto y Cadenas: invertir(String texto)	2
2.3	Métodos de Arrays: ordenar(int[] array)	2
2.4	Métodos de Validación y Autenticación: validarEmail(String email)	2
2.5	Métodos de Operaciones Matemáticas: raizCuadrada(double x)	2
2.6	Métodos Generales: aleatorio(int min, int max)	3
2.7	Capturas de pantalla métodos y pruebas unitarias	3
3	Compilación y Generación de JAR	3
3.1	Documentación con JavaDocs:.....	4
3.1.1	Ejemplo de código	4
3.1.2	Ejemplo de la documentación en java docs	6

Proyecto Junit

1 OBJETIVO

En este proyecto, nos hemos propuesto crear una librería que implemente seis métodos distintos, cada uno perteneciente a una categoría específica. Además, hemos integrado pruebas unitarias utilizando JUnit 5 para garantizar la robustez y fiabilidad de nuestra librería. Todo esto ha sido configurado dentro de un proyecto Gradle en IntelliJ para facilitar el desarrollo y la gestión del código.

2 CATEGORIZACIÓN DE MÉTODOS:

En este proyecto, se han implementado métodos en distintas categorías, los cuales se detallan a continuación:

2.1 MÉTODOS NUMÉRICOS: FACTORIAL(INT N)

Este método calcula el factorial de un número entero positivo. Se han realizado pruebas para verificar su correcto funcionamiento en diferentes escenarios, incluyendo números positivos, cero y números negativos.

2.2 MÉTODOS DE TEXTO Y CADENAS: INVERTIR(STRING TEXTO)

Este método invierte una cadena de texto. Se han realizado pruebas para asegurar que la inversión se realiza correctamente en diferentes casos, incluyendo cadenas no vacías, vacías y con espacios.

2.3 MÉTODOS DE ARRAYS: ORDENAR(INT[] ARRAY)

Este método ordena un array de enteros de menor a mayor. Se han realizado pruebas para verificar que el array se ordena correctamente en diferentes situaciones, incluyendo arrays desordenados, ordenados, vacíos y de un solo elemento.

2.4 MÉTODOS DE VALIDACIÓN Y AUTENTICACIÓN: VALIDAREMAIL(STRING EMAIL)

Este método verifica si una cadena de texto representa una dirección de correo electrónico válida. Se han realizado pruebas para asegurar que el método valida correctamente distintos formatos de direcciones de correo electrónico, incluyendo direcciones válidas e inválidas.

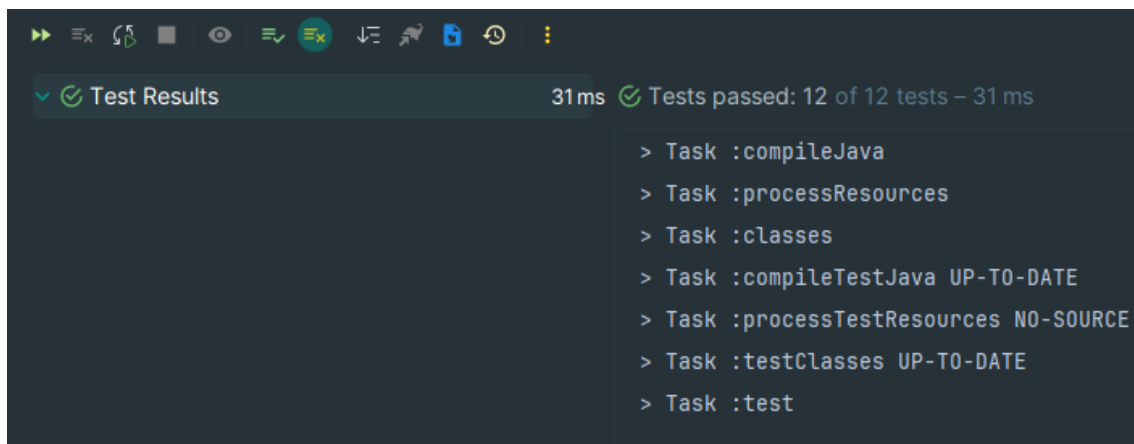
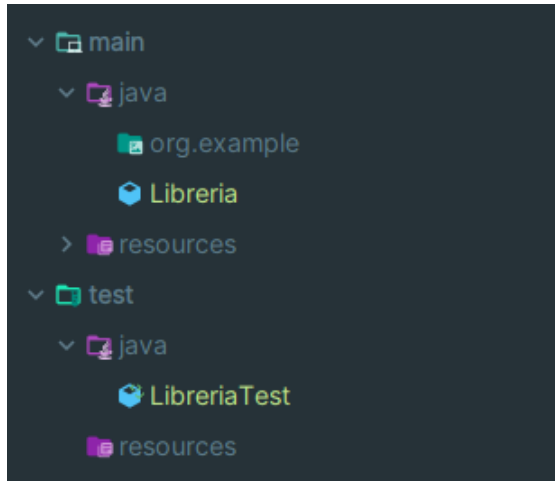
2.5 MÉTODOS DE OPERACIONES MATEMÁTICAS: RAIZCUADRADA(DOUBLE X)

Este método calcula la raíz cuadrada de un número real. Se han realizado pruebas para verificar que el cálculo se realiza correctamente en diferentes situaciones, incluyendo números positivos, cero y números negativos.

2.6 MÉTODOS GENERALES: ALEATORIO(INT MIN, INT MAX)

Este método genera un número aleatorio entre dos límites. Se han realizado pruebas para asegurar que el método genera un número dentro del rango especificado y que lanza una excepción cuando el límite inferior es mayor que el límite superior.

2.7 CAPTURAS DE PANTALLA MÉTODOS Y PRUEBAS UNITARIAS



3 COMPILACIÓN Y GENERACIÓN DE JAR

El proyecto ha sido compilado y se ha generado un archivo JAR que contiene la librería junto con las pruebas unitarias. Este archivo JAR proporciona una forma conveniente de distribuir la librería y asegura la portabilidad y la consistencia en su uso en diferentes entornos de desarrollo y ejecución.

Este proceso de compilación y generación de JAR asegura que la librería y las pruebas unitarias estén disponibles como un conjunto coherente y listo para su uso en cualquier proyecto Java que requiera su funcionalidad.

Lo hacemos entrando dentro de las herramientas de IntelliJ, en Project structure creamos un .jar y le hacemos un build con artifacts.

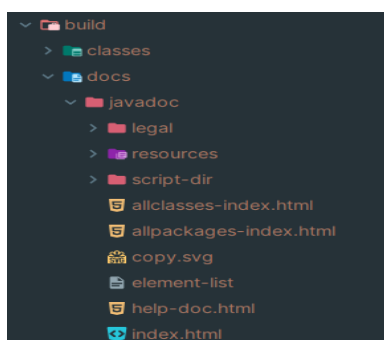


3.1 DOCUMENTACIÓN CON JAVADocs:

Para documentar el proyecto con JavaDocs, se ha utilizado comentarios especiales en el código fuente siguiendo la convención de JavaDocs. Estos comentarios están ubicados encima de cada método y proporcionan una descripción detallada de su funcionalidad, parámetros, valor de retorno y posibles excepciones. Además, se han etiquetado los comentarios con las etiquetas @param, @return y @throws para proporcionar información estructurada sobre los métodos.

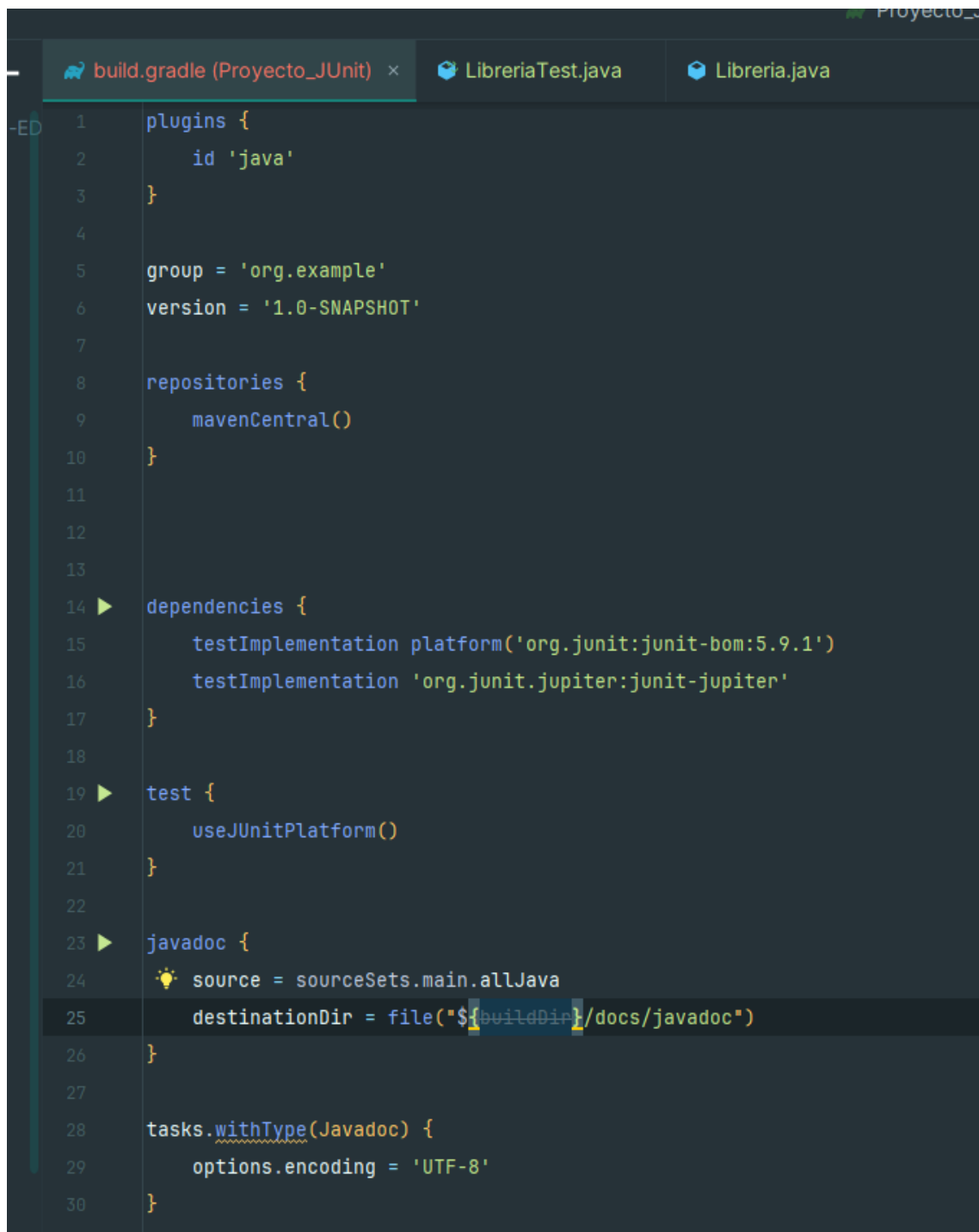
3.1.1 Ejemplo de código

```
/**
 * Método de la categoría Métodos Numéricos
 * Calcula el factorial de un número entero positivo.
 * @author Andreu Orenga
 * @param n el número del que se calculará el factorial
 * @return el factorial de n
 * @throws IllegalArgumentException si n es negativo
 */
```



Para generar el archivo HTML de la documentación JavaDocs utilizando Gradle, se ha configurado un script de Gradle que incluye la tarea javadoc. Esta tarea utiliza la herramienta javadoc de Java para generar la documentación a partir de los comentarios en el código fuente y la guarda en formato HTML en una carpeta específica del proyecto.

El script de Gradle para generar la documentación JavaDocs y el archivo HTML se vería así:



```
1 plugins {
2     id 'java'
3 }
4
5 group = 'org.example'
6 version = '1.0-SNAPSHOT'
7
8 repositories {
9     mavenCentral()
10 }
11
12
13
14 dependencies {
15     testImplementation platform('org.junit:junit-bom:5.9.1')
16     testImplementation 'org.junit.jupiter:junit-jupiter'
17 }
18
19 test {
20     useJUnitPlatform()
21 }
22
23 javadoc {
24     source = sourceSets.main.allJava
25     destinationDir = file("${buildDir}/docs/javadoc")
26 }
27
28 tasks.withType(Javadoc) {
29     options.encoding = 'UTF-8'
30 }
```

3.1.2 Ejemplo de la documentación en java docs

PACKAGE

CLASS

TREE

INDEX

HELP

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

Unnamed Package

Classes

Class	Description
Libreria	

Class Libreria

java.lang.Object[Ⓢ]

Libreria

public class Libreria

extends Object[Ⓢ]

Constructor Summary

Constructors

Constructor	Description
Libreria()	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static int	aleatorio(int min, int max)	Genera un número aleatorio entre dos límites.
static int	factorial(int n)	Método de la categoría Métodos Numéricos Calcula el factorial de un número entero positivo.
static String [Ⓢ]	invertir(String[Ⓢ] texto)	Invierte una cadena de texto.
static void	ordenar(int[] array)	Ordena un array de enteros de menor a mayor.
static double	raizCuadrada(double x)	Calcula la raíz cuadrada de un número real.
static boolean	validarEmail(String[Ⓢ] email)	Comprueba si una cadena de texto es un email válido.

Methods inherited from class java.lang.Object[Ⓢ]

clone[Ⓢ], equals[Ⓢ], finalize[Ⓢ], getClass[Ⓢ], hashCode[Ⓢ], notify[Ⓢ], notifyAll[Ⓢ], toString[Ⓢ], wait[Ⓢ], wait[Ⓢ], wait[Ⓢ]