

# Evaluación de expresiones matemáticas en Java

---

Evaluar expresiones matemáticas es útil en proyectos en los que queremos calcular expresiones matemáticas proporcionadas en formato de cadena (String).

Se puede emplear la biblioteca de *Javascript* de Java, incorporada en la API o bibliotecas de terceros.

## JAVA SCRIPTING API

Usando la API de **JavaScript** integrada en el JDK permite ejecutar secuencias de comandos pequeña de un modo potente sin uso de bibliotecas de terceros.

Todas las clases e interfaces de esta API están en el paquete **javax.script**.

La PI contiene las interfaces **ScriptEngineManager** y **ScriptEngine** que nos permiten evaluar JavaScript. Antes de Java 8, Java venía con el motor *Rhino*. Sin embargo, desde Java 8 en adelante, Java viene con el **motor Nashorn más nuevo y más potente**.

---

### Obtener una instancia de ScriptEngine

Para crear un *ScriptEngine*, primero tenemos que crear una instancia de **ScriptEngineManager**. Una vez que tengamos la instancia, debemos llamar al método **ScriptEngineManager#getEngineByName** para obtener el *ScriptEngine*:

```
ScriptEngineManager scriptEngineManager
    = new ScriptEngineManager();
ScriptEngine scriptEngine
    = scriptEngineManager.getEngineByName("JavaScript");
```

---

### Evaluación de expresiones simples

Podemos usar la instancia de *scriptEngine* anterior para llamar al método **ScriptEngine#eval**:

```
String expresion = "3+2";
Integer resultado = (Integer) scriptEngine.eval(expresion);
```

---

## Evaluación de expresiones con variables

Dado que estamos utilizando el motor de JavaScript, **podemos agregar variables directamente a las expresiones como lo hacemos en JavaScript.**

```
String expresion = "x=2; y=3; 3*x+2*y;";  
Double resultado = (Double) scriptEngine.eval(expresion);
```

JavaScript no tiene métodos directos para realizar operaciones matemáticas y requiere acceso al objeto Math. Por lo tanto, no podemos resolver expresiones matemáticas utilizando la API de Javascript.

### EXP4J

[exp4j](#) es una biblioteca de código abierto que se puede utilizar para evaluar expresiones y funciones matemáticas. La biblioteca implementa [el algoritmo Shunting Yard de Dijkstra](#), un método para analizar expresiones matemáticas especificadas en [notación infija](#).

Además de utilizar operadores y funciones estándar, exp4j nos permite crear operadores y funciones personalizados.

---

## Dependencias Maven

```
<dependency>  
  <groupId>net.objecthunter</groupId>  
  <artifactId>exp4j</artifactId>  
  <version>0.4.8</version>  
</dependency>
```

---

## Evaluación de expresiones simples

```
Expression expresion = new ExpressionBuilder("9+9").build();  
double resultado = expresion.evaluate();
```

---

## Uso de variables

```
Expression expresion = new ExpressionBuilder("3x+2y")  
    .variables("x", "y")  
    .build()  
    .setVariable("x", 2)  
    .setVariable("y", 3);  
double resultado = expresion.evaluate();
```

---

## Evaluación de funciones matemáticas

```
Expression expresion
    = new ExpressionBuilder("sin(x)*sin(x)+cos(x)*cos(x)+1")
    .variables("x")
    .build()
    .setVariable("x", 0.5);
double resultado = expresion.evaluate();
```

## JAVALUATOR

[Javaluator](#) es otra biblioteca independiente y ligera que está disponible de forma gratuita. Al igual que exp4j, *Javaluator* también se usa para evaluar expresiones infijas.

---

## Dependencias Maven

```
<dependency>
    <groupId>com.fathzer</groupId>
    <artifactId>javaluator</artifactId>
    <version>3.0.3</version>
</dependency>
```

---

## Avaluación de expresiones simples

```
String expresion = "3+2";
DoubleEvaluator eval = new DoubleEvaluator();
Double resultado = eval.evaluate(expresion);
```

---

## Evaluación de expresiones que contienen variables

```
String expresion = "3*x+2*y";
DoubleEvaluator eval = new DoubleEvaluator();
StaticVariableSet<Double> variables
    = new StaticVariableSet<Double>();
variables.set("x", 2.0);
variables.set("y", 3.0);
Double resultado = eval.evaluate(expresion, variables);
```

---

## Evaluación de funciones matemáticas

```
String expresion = "sin(x)*sin(x)+cos(x)*cos(x)";  
DoubleEvaluator eval = new DoubleEvaluator();  
StaticVariableSet<Double> variables  
    = new StaticVariableSet<Double>();  
variables.set("x", 0.5);  
Double resultado = eval.evaluate(expresion, variables);
```