

# **RESULTADOS CALCULADORA**

## PruebaSuma.feature

```
Scenario: Suma estandar # src/test/resources/calculator/PruebaSuma.feature:5
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Sumo 2 y 3 # calculator.StepDefinitions.suma(double,double)
  Then El resultado de la suma es 5 # calculator.StepDefinitions.resultadoSum(double)
```

```
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0,526s
```

## PruebaResta.feature

```
Scenario: Resta estandar # src/test/resources/calculator/PruebaResta.feature:5
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Resto 5 y 3 # calculator.StepDefinitions.resta(double,double)
  Then El resultado de la resta es 2 # calculator.StepDefinitions.resultadosRest(double)
```

```
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0,561s
```

## PruebaMult.feature

```
Scenario: Mult estandar # src/test/resources/calculator/PruebaMult.feature:5
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Multiplico 5 y 7 # calculator.StepDefinitions.multi(double,double)
  Then El resultado de mult es 35 # calculator.StepDefinitions.resultadosMult(double)
```

```
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0,596s
```

## PruebaDivision.feature

```
Scenario Outline: DivisiÃ³n estandar # src/test/resources/calculator/PruebaDivision.feature:12
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Divido 10 y 2 # calculator.StepDefinitions.division(double,double)
  Then El resultado de la divisiÃ³n es 5 # calculator.StepDefinitions.resultadosDivid(double)
```

```
Scenario Outline: DivisiÃ³n estandar # src/test/resources/calculator/PruebaDivision.feature:13
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Divido 6 y 6 # calculator.StepDefinitions.division(double,double)
  Then El resultado de la divisiÃ³n es 1 # calculator.StepDefinitions.resultadosDivid(double)
```

```
Scenario: DivisiÃ³n de valores negativos # src/test/resources/calculator/PruebaDivision.feature:15
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Divido -10 y -5 # calculator.StepDefinitions.division(double,double)
  Then El resultado de la divisiÃ³n es 2 # calculator.StepDefinitions.resultadosDivid(double)
```

```
3 Scenarios (3 passed)
9 Steps (9 passed)
0m0,652s
```

## PruebaFactorial.feature

```
Scenario: Factorial de valores # src/test/resources/calculator/PruebaFactorial.feature:5
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Hago el factorial de 3 # calculator.StepDefinitions.factorial(int)
  Then El factorial es 6 # calculator.StepDefinitions.resultadosFact(int)
```

```
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0,597s
```

## PruebaPrimo.feature

```
Scenario: ¿Este valor es primo? # src/test/resources/calculator/PruebaPrimo.feature:5
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Se comprueba si 13 es primo # calculator.StepDefinitions.primo(int)
  Then El resultado es "true" # calculator.StepDefinitions.resultadosPrimo(java.lang.String)

Scenario Outline: ¿Estos valores son primos? # src/test/resources/calculator/PruebaPrimo.feature:17
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Se comprueba si 5 es primo # calculator.StepDefinitions.primo(int)
  Then El resultado es "true" # calculator.StepDefinitions.resultadosPrimo(java.lang.String)

Scenario Outline: ¿Estos valores son primos? # src/test/resources/calculator/PruebaPrimo.feature:18
  Given Tengo una Calculadora # calculator.StepDefinitions.intializeCalculator()
  When Se comprueba si 6 es primo # calculator.StepDefinitions.primo(int)
  Then El resultado es "false" # calculator.StepDefinitions.resultadosPrimo(java.lang.String)

3 Scenarios (3 passed)
9 Steps (9 passed)
0m0,642s
```