

Tercero: Completa las pruebas con JUnit

Errores del test para el método factorial()

1. En el caso $n = 20$
 - El resultado obtenido es diferente, ha sido -2102132736.
 - El motivo es que se encuentra fuera de rango para un int, ha desbordado.
 - Se soluciona cambiando el tipo de variable a long de la siguiente forma:

```
public static long factorial(int n) {  
    long resultado;  
  
    resultado = 1;  
    for (int i = 2; i <= n; i++) {  
        resultado = resultado * i;  
    }  
    return resultado;  
}
```

2. En los casos $n = -3$ y $n = -1$
 - Ocurre una anomalía, el resultado no devuelve una excepción.
 - El motivo es que por defecto, el método devuelve un 1 para números menores a 2.
 - Se soluciona añadiendo una nueva condición para que se genere una excepción al introducir números negativos de la siguiente forma:

```
public static long factorial(int n) {  
    long resultado;  
  
    if(n < 0) {  
        throw new ArithmeticException("No se puede calcular el  
factorial de un número negativo");  
    }  
  
    resultado = 1;  
    for (int i = 2; i <= n; i++) {  
        resultado = resultado * i;  
    }  
    return resultado;  
}
```

Errores del test para el método divisible()

1. En el caso multiplo = 18, divisor = 0
 - El resultado generado ha sido ArithmeticException.
 - El motivo es que no se puede dividir por 0.
 - Se soluciona controlando la excepción de la siguiente manera:

```
public static boolean divisible(int multiplo, int divisor) {  
    boolean resultado;  
  
    if (divisor == 0) {  
        resultado = false;  
    } else if (multiplo % divisor == 0) {  
        resultado = true;  
    } else {  
        resultado = false;  
    }  
  
    return resultado;  
}
```