# **Método: Factorial.**

**Casos de prueba Resultado esperado**

n = -3 Excepción

n = -1 Excepción

Para los 2 casos da el mismo error: Expected java.lang.IllegalArgumentException to be thrown, but nothing was thrown.

Para arreglar el problema, se tiene que añadir el siguiente código antes del bucle for:

if (n < 0) { //Añado este if para la excepción de número negativo

throw new IllegalArgumentException("El número debe de ser un entero no negativo");

}

**Casos de prueba Resultado esperado**

n = 20 2432902008176640000

Causa error por estar fuera de rango.

Para arreglar el problema, modificamos el código para que admita variables tipo long y acepte el resultado:

public static long factorial(long n) {

long resultado;

También modificamos el test añadiendo una L al número para que se marque como tipo long:

*assertEquals*(2432902008176640000L, Enteros.*factorial*(20)); //L para especificar que es long

# **Método: Divisible.**

**Casos de prueba Resultado esperado**

multiplo = 18 y divisor = 0 false

Da error al tener el divisor con valor 0.

Tenemos que añadir al test la línea:

*assertThrows*(ArithmeticException.class, () -> Enteros.*divisible*(18, 0));