

C.F.G.S.: DESARROLLO DE APLICACIONES WEB  
Módulo: Programación

## APÉNDICE

### "Las pequeñas cosas llevan a grandes transformaciones"

#### ASERCIONES

En Java, las **aserciones (assertions)** se utilizan para realizar comprobaciones durante la ejecución del programa. Sirven para verificar si una condición lógica es verdadera en un punto determinado del código. Si la condición no se cumple, el programa lanza un error de tipo **AssertionError**, si se cumple el programa continúa como si nada.

Las aserciones son útiles para realizar **depuración y validación interna** durante el desarrollo, pero no se utilizan en código de producción.

Sintaxis:

assert condición: "mensaje";

Si la condición no se cumple el programa termina y se muestra el mensaje.

Ejemplo:

```
public class Factorial {
    public static void main(String[] args) {
        int numero = -5;
        int resultado = calcularFactorial(numero);
        System.out.println("El factorial de " + numero + " es " + resultado);
    }

    /**
     * Calcula el factorial de un número, si es negativo y están activadas las aserciones
     * muestra un mensaje y termina
     * @param n
     * @return
     */
    public static int calcularFactorial(int n) {
        // Aseguramos que el número sea mayor o igual a cero
        assert n >= 0 : "El número debe ser mayor o igual a cero, pero fue " + n;

        int factorial = 1;
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }
        return factorial;
    }
}
```

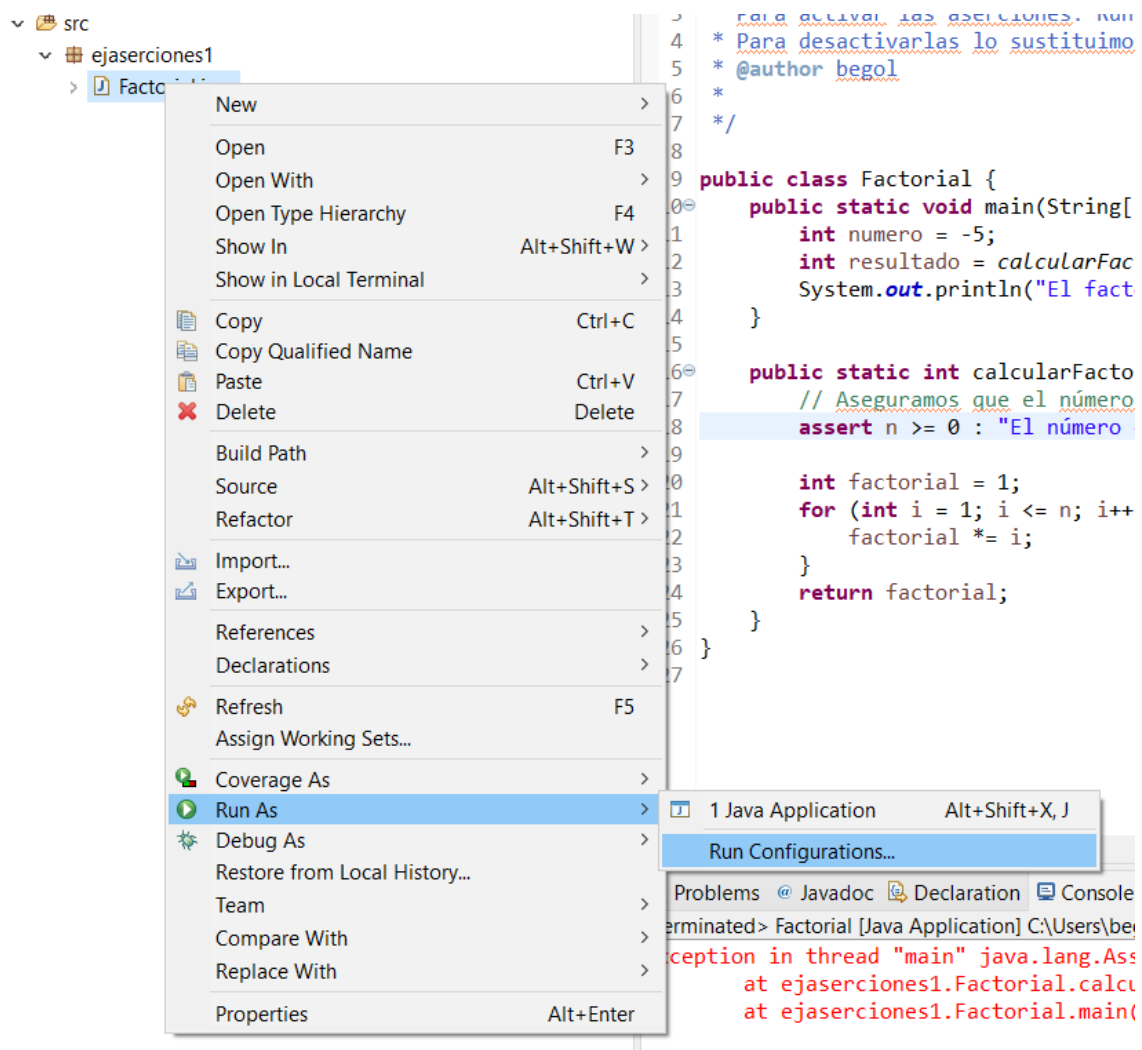
En este caso entramos en la función con un valor negativo. El programa termina y se muestra el siguiente mensaje:

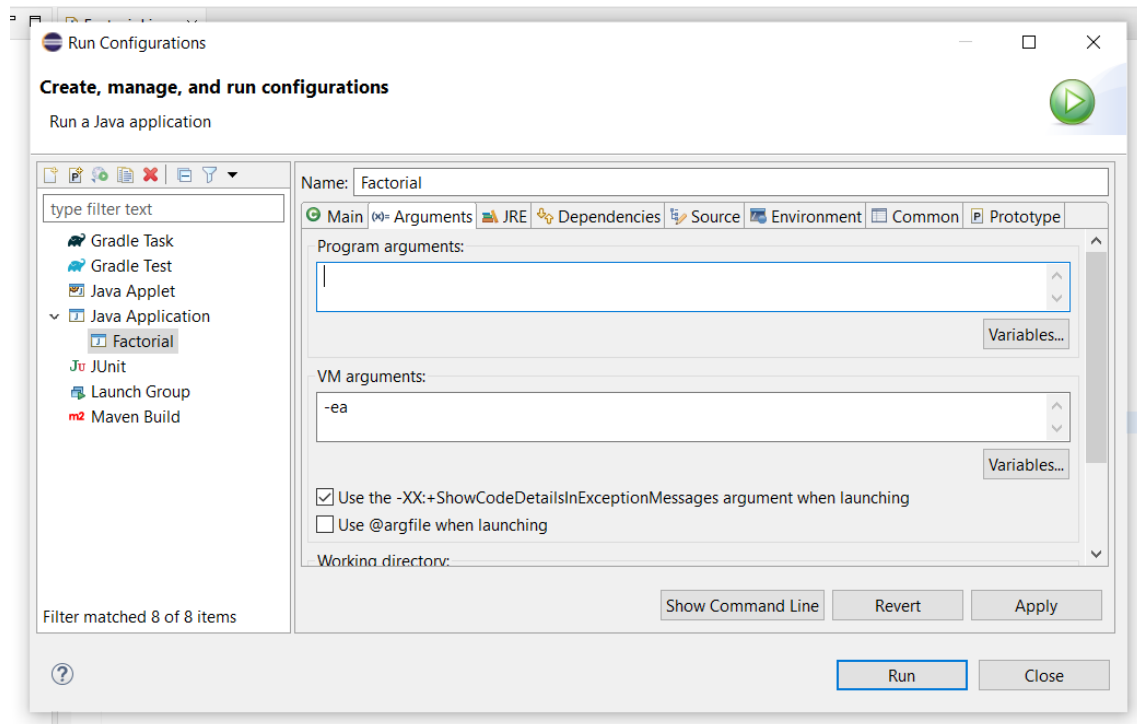
```
Exception in thread "main" java.lang.AssertionError: El número debe ser mayor o igual a cero, pero fue -5
    at ejaserciones1.Factorial.calcularFactorial(Factorial.java:18)
    at ejaserciones1.Factorial.main(Factorial.java:12)
```

Por defecto las aserciones están desactivadas en java. Para activarlas hay que entrar en:

Run As > Run Configurations > Arguments > VM Arguments

y poner la directiva -ea





Para desactivarlas, la directiva es -da

Esta directiva puede funcionar a nivel de archivo, o proyecto.

### PARÁMETROS EN EL MAIN

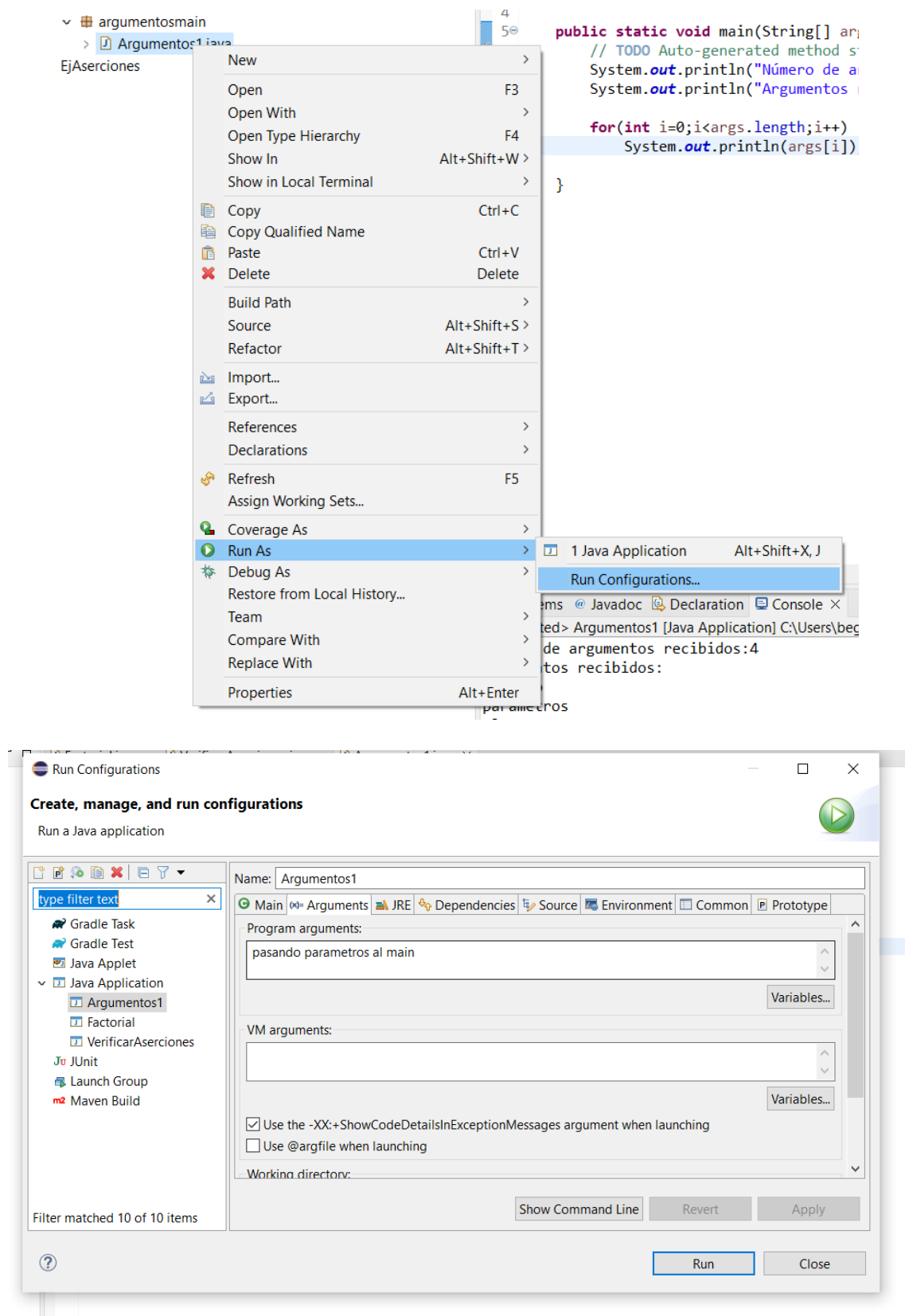
Se pueden pasar parámetros a través del main en un programa java. Estos parámetros son recogidos por el programa como un array de String. Se deben introducir separados por espacios.

```
public class Argumentos1 {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Número de argumentos recibidos:"+ args.length);  
        System.out.println("Argumentos recibidos:");  
  
        for(int i=0;i<args.length;i++)  
            System.out.println(args[i]);  
    }  
}
```

Para introducir los datos en eclipse

Run As > Run Configurations > Arguments > Program Arguments

E introducimos todos los valores separados por espacios en blanco.



Al ejecutar el programa esta sería la salida:

```
<terminated> Argumentos1 [Java Application]
Número de argumentos recibidos:4
Argumentos recibidos:
pasando
parametros
al
main
```