

# ENTORNOS DE DESARROLLO

## TEMA 3: DEPURACIÓN Y REALIZACIÓN DE PRUEBAS

# 1. Práctica 1

2

- La finalidad de esta practica era que aprendieseis a realizar pruebas de caja negra sobre la clase Factorial.

```
public class Factorial {  
    public static int factorial (int x) throws Exception{  
        int response = 1;  
  
        if(x<0){  
            throw new Exception();  
        }  
        if(x>0){  
            response = x * factorial(x-1);  
        }  
        return response;  
    }  
}
```

# 1. Práctica 1

3

- El **factorial** de un entero positivo  $n$ , el **factorial** de  $n$  o  $n$  **factorial** se define en principio como el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta  $n$ .  
Por ejemplo:  $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$ .

$$1! = 1$$

$$2! = 2(1) = 2$$

$$3! = 3(2)(1) = 6$$

$$4! = 4(3)(2)(1) = 24$$

$$5! = 5(4)(3)(2)(1) = 120$$

# 1. Práctica 1

4

- En la práctica, con los valores de la función factorial que os deje propuestos, los valores esperados no se correspondían con las salidas del programa.

```
public void testFactorial() throws Exception {  
    //fail("Not yet implemented");  
    double x = 0;  
    double y = 0;  
    double z = 0;  
    assertEquals(1, Factorial.factorial(0),0);  
    y = Factorial.factorial(0);  
    assertEquals(24, Factorial.factorial(4),0);  
    y = Factorial.factorial(4);  
    assertEquals(120, Factorial.factorial(5),0);  
    z = Factorial.factorial(5);  
  
    boolean excepcionLanzada = false;  
}
```

finished after 0,034 seconds

Runs: 1/1   Errors: 0   Failures: 1

FactorialTest [Runner: JUnit 4] (0,003 s)

testFactorial (0,003 s)

Failure Trace

java.lang.AssertionError: expected:<24.0> but was:<11.0>  
at FactorialTest.testFactorial(FactorialTest.java:16)

# 1. Práctica 1

5

- Lo cual, indica:
  - Si conocemos el dato correcto de salida para dicha entrada de datos en la función factorial, esa fallando nuestra función factorial, ya que es la función que estamos comprobando.
  - El programa nos devuelve que el factorial de 4 es 11. ¿Porque? → La función esta realizando la siguiente operación:  $4! = 4 + 3 + 2 + 1 + 1 = 11$
  - $(4 + \text{factorial}(3)) + (3 + \text{factorial}(2)) + (2 + \text{factorial}(1)) + (1 + \text{factorial}(0))$

# 1. Práctica 1

6

- Realmente conocemos que el  $\text{factorial}(4) = 24$
- Hay que encontrar el error en la función factorial  
→ El error es el signo de la operación:  $4! = 4 * 3 * 2 * 1 = 24$
- $(4 + \text{factorial}(3)) * (3 + \text{factorial}(2)) * (2 + \text{factorial}(1)) * (1 + \text{factorial}(0))$

```
public class Factorial {  
    public static int factorial (int x) throws Exception{  
        int response = 1;  
  
        if(x<0){  
            throw new Exception();  
        }  
        if(x>0){  
            response = x + factorial(x-1);  
        }  
        return response;  
    }  
}
```