

# Planificación y realización de pruebas

## Pruebas estructurales o de caja blanca

Esta práctica ha sido desarrollada en su totalidad por Román Ginés Martínez Ferrández (rgmf@riseup.net) salvo referencias al pie de página.

Todas las imágenes y todos los recursos utilizados son de Dominio Público a menos que se diga lo contrario.



Creative Commons Reconocimiento - NoComercial - CompartirIgual  
CC by-nc-sa

### Pregunta 1

En las especificaciones de un programa se indica que se tiene que desarrollar un método en Java que compare dos números e indique qué número es mayor que otro o si son iguales. Las salidas tienen que seguir este formato:

- El número 5 es mayor que el número 3
- Los números son iguales

Se ha desarrollado el método y, aquí, te muestro el resultado. Pude tener fallos, eso no debe preocuparnos porque lo único que tenemos que hacer nosotros es planificar las pruebas que se tienen que pasar.

```
public void comparaNumeros(int num1, int num2) {  
    if (num1 > num2) {  
        System.out.println("El número " + num1 + " es mayor que el número " + num2);  
    } else if (num2 > num1) {  
        System.out.println("El número " + num2 + " es mayor que el número " + num1);  
    } else {  
        System.out.println("Los numeros " + num1 + " y " + num2 + " son iguales");  
    }  
}
```

¿Cuántas sentencias hay en el cuerpo de este método?

5 sentencias.

Enumera los casos de prueba que realizarías (indicando entradas y salida esperada). Recuerda que tienes que planificar estas pruebas en base a las especificaciones y no en base al método desarrollado:

Entradas	Salida
Num1 = 5, num2 = 3	El número 5 es mayor que el número 3
Num1 = 3, num2 = 5	El número 5 es mayor que el número 3
Num1 = 5, num2 = 5	Los número 5 y 5 son iguales

## Pregunta 2

En las especificaciones de un programa se indica que se tiene que desarrollar un método en Java que reciba una cadena de texto y devuelva el número de vocales que hay en dicha cadena. Ten en cuenta que:

- La cadena podría ser nula o podría estar vacía
- Las vocales podrían estar en mayúscula o minúscula

Se ha desarrollado el método y, aquí, te muestro el resultado. Pude tener fallos, eso no debe preocuparnos porque lo único que tenemos que hacer nosotros es planificar las pruebas que se tienen que pasar.

```
public int numeroVocales(String cadena) {  
    int contador = 0;  
  
    for (int i = 0; i < cadena.length(); i++) {  
        if(cadena.charAt(i)=='a' || cadena.charAt(i)=='e' || cadena.charAt(i)=='i' ||  
            cadena.charAt(i)=='o' || cadena.charAt(i)=='u') {  
            contador++;  
        }  
    }  
  
    return contador;  
}
```

¿Cuántas sentencias hay en el cuerpo de este método?

Hay 11 sentencias.

Enumera los casos de prueba que realizarías (indicando entradas y salida esperada). Recuerda que tienes que planificar estas pruebas en base a las especificaciones y no en base al método desarrollado:

Entradas	Salida
cadena=null	Excepción
cadena="hola"	Contador=2
cadena="HOLA"	Contador=2
cadena="nm"	Contador=0
cadena="NM"	Contador=0
Cadena=""	Contador=0

## Pregunta 3

En las especificaciones de un programa se indica que se tiene que desarrollar un método en Java que sume los números que hay en el rango indicado, ambos incluidos. Si el rango es erróneo (el primer valor es mayor que el segundo) se devolverá 0. Si los límites son iguales, también se devolverá 0.

Se ha desarrollado el método y, aquí, te muestro el resultado. Pude tener fallos, eso no debe preocuparnos porque lo único que tenemos que hacer nosotros es planificar las pruebas que se tienen que pasar.

```
public int sumaNumeros(int desde, int hasta) {  
    int suma = 0;  
  
    if (desde > hasta) {  
        return suma;  
    }  
  
    for (int i = desde; i <= hasta; i++) {  
        suma += i;  
    }  
  
    return suma;  
}
```

¿Cuántas sentencias hay en el cuerpo de este método?

Hay 8 sentencias.

Enumera los casos de prueba que realizarías (indicando entradas y salida esperada). Recuerda que tienes que planificar estas pruebas en base a las especificaciones y no en base al método desarrollado:

Entradas	Salida
Desde=3, hasta=1	Suma=0
Hasta=3, desde=1	Suma=6
Hasta=1, desde=1	suma=0

## Pregunta 4

En las especificaciones de un programa se indica que se tiene que desarrollar un método en Java que dado un día de la semana y un código de idioma, devuelva el día en texto.

En cuanto al código de idioma, se acepta solo “en” para inglés. En cualquier otro caso se devolverá el día de la semana en español.

Si el número de día está fuera del rango [1, 7], entonces se devolverá el mensaje: “Día desconocido”.

Se ha desarrollado el método y, aquí, te muestro el resultado. Pude tener fallos, eso no debe preocuparnos porque lo único que tenemos que hacer nosotros es planificar las pruebas que se tienen que pasar.

```
public String diaSemana(int dia, String idioma) {
    if (dia >= 1 && dia <= 7) {
        switch (dia) {
            case 1:
                return idioma.equals("en") ? return "Monday" : "Lunes";
                break;
            case 2:
                return idioma.equals("en") ? return "Tuesday" : "Martes";
                break;
            case 3:
                return idioma.equals("en") ? return "Wednesday" : "Miércoles";
                break;
            case 4:
                return idioma.equals("en") ? return "Thursday" : "Jueves";
                break;
            case 5:
                return idioma.equals("en") ? return "Friday" : "Viernes";
                break;
            case 6:
                return idioma.equals("en") ? return "Saturday" : "Sábado";
                break;
            case 7:
                return idioma.equals("en") ? return "Sunday" : "Domingo";
                break;
        }
    } else {
        return "Día desconocido";
    }
}
```

Enumera los casos de prueba que realizarías (indicando entradas y salida esperada). Recuerda que tienes que planificar estas pruebas en base a las especificaciones y no en base al método desarrollado:

Entradas	Salida
dia=0, idioma=""	Día desconocido
dia=8, idioma=""	Día desconocido
dia=1, idioma=""	Lunes
dia=2, idioma=""	Martes

dia=3, idioma=""	Miércoles
dia=4, idioma=""	Jueves
dia=5, idioma=""	Viernes
dia=6, idioma=""	Sábado
dia=7, idioma=""	Domingo
dia=1, idioma="en"	Monday
dia=2, idioma="en"	Tuesday
dia=3, idioma="en"	Wednesday
dia=4, idioma="en"	Thursday
dia=5, idioma="en"	Friday
dia=6, idioma="en"	Saturday
dia=7, idioma="en"	Sunday

## Pregunta 5

En las especificaciones de un programa se indica que se tiene que desarrollar un método en Java que busque en un array de números enteros el número indicado como segundo parámetro.

Este método devolverá el número de veces que aparece dicho número.

Se ha desarrollado el método y, aquí, te muestro el resultado. Pude tener fallos, eso no debe preocuparnos porque lo único que tenemos que hacer nosotros es planificar las pruebas que se tienen que pasar.

```
public boolean busca(int tabla[], int numero) {  
    int i = 0;  
    boolean esta;  
  
    while (i < tabla.length && t[i] != numero) {  
        i++;  
    }  
  
    if (i < tabla.length) {  
        esta = true;  
    } else {  
        esta = false;  
    }  
  
    return esta;  
}
```

¿Cuántas sentencias hay en este programa?

Hay 9 sentencias.

Enumera los casos de prueba que realizarías (indicando entradas y salida esperada). Recuerda que tienes que planificar estas pruebas en base a las especificaciones y no en base al método desarrollado:

Entradas	Salida
tabla[]={}	esta=false, nº veces=""
tabla[]={1}, numero=1	esta=true, nº veces="1"
tabla[]={1, 2}, numero=3	esta=false, nº veces=""
tabla[]={4, 4, 4, 4}, numero=4	esta=true, nº veces="4"
tabla[]={1, 2, 3, 4}, numero=1	esta=true, nº veces="1"