# EJERCICIO2 Pruebas Código

A partir del siguiente código de un método **(obtenerNumeros)** que recibe como parámetro un valor de tipo entero y una cadena de manera que si la cadena es igual a "par" el método devuelve todos los números pares entre 1 y el valor entero introducido y si la cadena es igual a "impar" el método devuelve todos los números impares entre 1 y el valor entero introducido, realiza las pruebas de cubrimiento

* Función
* Sentencias
* Decisión
* Condición

**import java.util.ArrayList;**

**import java.util.List;**

**public class ParImpar {**

**public static void main(String[] args) {**

**int valor = 10; // El valor entero introducido**

**String tipo = "par"; // La cadena ("par" o "impar")**

**List<Integer> numeros = obtenerNumeros(valor, tipo);**

**// Imprimir los números obtenidos**

**System.out.println("Números " + tipo + "es entre 1 y " + valor + ":");**

**for (int num : numeros) {**

**System.out.print(num + " ");**

**}**

**}**

**public static List<Integer> obtenerNumeros(int valor, String tipo) {**

**List<Integer> numeros = new ArrayList<>();**

**if (tipo.equals("par")) {**

**for (int i = 2; i <= valor; i += 2) {**

**numeros.add(i);**

**}**

**} else if (tipo.equals("impar")) {**

**for (int i = 1; i <= valor; i += 2) {**

**numeros.add(i);**

**}**

**}**

**return numeros;**

**}**

**}**

|  |  |  |
| --- | --- | --- |
| **CUBRIMIENTO** | **PRUEBA** | **SATISFECHO** |
| Función |  | El método es llamado desde el método principal (main) de la clase **ParImpar** |
| Sentencias | (50,”par”) | Cubre todas las sentencias menos el **else if(tipo.equals(“impar”)){…}** |
| (50,”impar”) | Con este parámetro se cubre la sentencia **else if(tipo.equals(“impar”)){…}** |
| Decisión | (50,”par”) | Con estos valores se cubre la decisión verdadera del primer **if** y la falsa del segundo **if** |
| (50,”impar”) | Se cubre la decisión falsa del primer **if** y verdadera del segundo **if** |
| Condición | (50,”par”) | Se cubre la condición tipo.equals(“par”) pero no la condición tipo.equals(“impar”) |
| (50,”impar”) | Se cubre la condición tipo.equals(“impar”) pero no la condición tipo.equals(“par”) |