

Actividad Complejidad Ciclomática

Decimos que un número es perfecto si la suma de los divisores de ese número (exceptuándose él mismo) es igual al propio número. Por ejemplo:

- El 6 es un número perfecto ya que la suma de sus divisores (1+2+3) es 6.
- El 28 es un número perfecto ya que la suma de sus divisores (1+2+4+7+14) es 28.

```
13 public boolean esPerfecto(int j)
14 {
15     int suma;
16     boolean resultado;
17     int i;
18
19     suma=0;
20     resultado=false;
21     // buscamos todos los divisores de número
22     for (i=1; i<j ;i++)
23     {
24         if (j%i == 0)
25         { // i es divisor de numero
26             suma = suma + i;
27         } /* if */
28     } /* for i */
29     if (suma==j)
30     {
31         resultado=true;
32     } /* if */
33     return resultado;
34 } // esPerfecto
```

El anterior fragmento de código recibe como entrada un número j y retorna true si j es perfecto y false en caso contrario:

Para este fragmento de código se pide:

1. Diseñar pruebas de caja negra mediante una de las técnicas estudiadas, e implementarlas mediante JUnit.
2. Diseñar pruebas de caja blanca mediante la técnica de cobertura de caminos para lo que se hará:
 - 2.1. Representar el grafo.
 - 2.2. Determinar la complejidad ciclomática del programa usando las distintas técnicas descritas.
 - 2.3. Determinar los caminos básicos.

Si finalizas la actividad, puedes probar a repetirla con alguno de los ejercicios del módulo de programación que tengas resueltos.