

Pt2. Tests en JUnit i TDD

Actividad num 42 (de decimal a binario).

```
package com.iesebre;  
import java.util.Scanner;  
  
class Main {  
  
    public static void main(String[] args) {  
        int[] decimales = { 0, 2, 5, 50, 120, 1023, 51, 52, 60, 100, 53, 54, 55, 56, 57, 16, 17 };  
        for (int decimal : decimales) {  
            System.out.println("Convirtiendo decimal a binario el número " + decimal);  
            System.out.println("Manual: " + convertirDecimalABinarioManual(decimal));  
            System.out.println("Usando funciones: " + convertirDecimalABinario(decimal));  
        }  
    }  
  
    public static String convertirDecimalABinario(long decimal) {  
        return Long.toBinaryString(decimal);  
    }  
  
    public static String convertirDecimalABinarioManual(long decimal) {  
        if (decimal <= 0) {  
            return "0";  
        }  
        StringBuilder binario = new StringBuilder();  
        while (decimal > 0) {  
            short residuo = (short) (decimal % 2);  
            decimal = decimal / 2;  
            // Insertar el dígito al inicio de la cadena  
            binario.insert(0, String.valueOf(residuo));  
        }  
        return binario.toString();  
    }  
}
```

En este caso lo he hecho de 2 maneras diferentes, la primera forma es usando el método que nos ofrece java para poder transformar de binario a decimal, la segunda hice un método manualmente que nos ofrece lo mismo, simplemente que más elaborado y hecho por mi.

```
public static void main(String[] args) {  
    int[] decimales = { 0, 2, 5, 50, 120, 1023, 51, 52, 60, 100, 53, 54, 55, 56, 57, 16, 17 };  
    for (int decimal : decimales) {  
        System.out.println("Convirtiendo decimal a binario el número " + decimal);  
        System.out.println("Manual: " + convertirDecimalABinarioManual(decimal));  
        System.out.println("Usando funciones: " + convertirDecimalABinario(decimal));  
    }  
}
```

En la parte superior lo intente hacer de manera que el numero lo pones tú por teclado, pero se me hizo imposible y no encontraba una manera de hacer, así que decidí guardar unos cuantos números en un array y así usarlos como prueba...

Fue lo más difícil de lograr y al final me agobié y lo hice así, por lo demás me resultó bastante sencillo y no tuve casi fallos, ya que el método que nos ofrece java es bastante sencillo de usar.

```
public static String convertirDecimalABinario(long decimal) {  
    return Long.toBinaryString(decimal);  
}
```

El primer método que usamos, es un método que nos ofrece el propio java, aparte de eso lo ponemos en la clase "long" para poder usar números grandes.

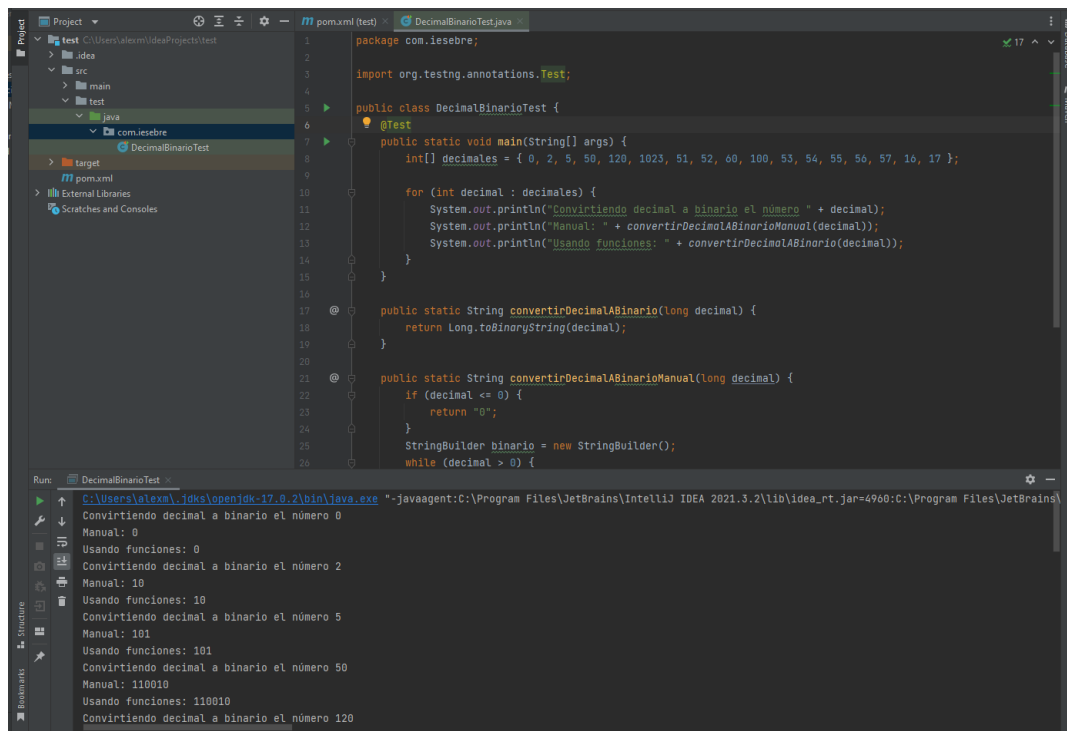
```
public static String convertirDecimalABinarioManual(long decimal) {  
    if (decimal <= 0) {  
        return "0";  
    }  
    StringBuilder binario = new StringBuilder();  
    while (decimal > 0) {  
        short residuo = (short) (decimal % 2);  
        decimal = decimal / 2;  
        // Insertar el dígito al inicio de la cadena  
        binario.insert(0, String.valueOf(residuo));  
    }  
    return binario.toString();  
}
```

Y el otro método es uno que he creado yo, simplemente usando la manera en que se transforma de decimal a binario de manera matemática (simplemente es ir dividiendo entre "2") también puse que si el número es inferior a 0 que me devuelve 0 para no tener números negativos.

Al ejecutar el programa nos sale tal que así:

```
C:\Users\alexm\jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea  
Convirtiendo decimal a binario el número 0  
Manual: 0  
Usando funciones: 0  
Convirtiendo decimal a binario el número 2  
Manual: 10  
Usando funciones: 10  
Convirtiendo decimal a binario el número 5  
Manual: 101  
Usando funciones: 101  
Convirtiendo decimal a binario el número 50  
Manual: 110010  
Usando funciones: 110010  
Convirtiendo decimal a binario el número 120  
Manual: 1111000
```

Como se puede apreciar, nos salen los 2 métodos, el manual y el propio de java..



Creamos un nuevo proyecto en xml para el apartado de tests...

Ponemos la característica de Test para poder ejecutarlo, creamos el apartado “@Test” y luego de eso le damos click derecho y run, y como se puede apreciar no nos das ningún error...

La verdad que tuve un par de problemas intentando crear este apartado, pero gracias a la comodidad de IntelliJ, prácticamente me lo ha solucionado el con sus consejos maestros.