

Callstack Decimal a binario

```
public String decimalABinario(int num){
    if (num < 2) {
        return Integer.toString(num);
    } else {
        return decimalABinario(num/2) + (num%2);
    }
}
```

~~recursividad.decABin(1), num=1~~
~~recursividad.decABin(2), num=2 → 1~~
~~recursividad.decABin(5), num=5 → 2~~
~~recursividad.decABin(10), num=10 → 5~~
~~recursividad.decABin(20), num=20 → 10~~
~~Main, num=20 ← ejemplo~~

Valor retornado

"10100"

Callstack Maximo Comun Divisor

~~MaxComDiv(36, 12), a: 36 b: 12~~ ^{condición (36%12)=0 ✓}
~~MaxComDiv(120, 36), a: 120 b: 36~~ ^{condición (120%36)=12 ✗} ~~res: (120%36)=12~~
~~MaxComDiv(156, 120), a: 156 b: 120~~ ^{condición (156%120)=36 ✗} ~~res: (156%120)=36~~
~~Main, a: 156, b: 120 ← ejemplo~~

Valor Retornado
12

residuo →

	1	3	3
156	120	36	12
	36	12	0

$\begin{array}{r} 1 \\ 120 \overline{) 156} \\ \underline{- 120} \\ 36 \end{array}$	$\begin{array}{r} 3 \\ 36 \overline{) 120} \\ \underline{- 108} \\ 12 \end{array}$	$\begin{array}{r} 3 \\ 12 \overline{) 36} \\ \underline{- 36} \\ 0 \end{array}$
---	--	---

```
public int MaxComDiv(int a, int b){
    if (a%b == 0){
        return b;
        // retornamos b ya que segun e
        // lo cual en el else guardamos
    } else {
        int res = a % b;
        a = b;
        b = res;
        return MaxComDiv(a, b);
    }
}
```

Callstack fibonacci

```
public int fib(int n){  
    if (n == 0 || n == 1){  
        return n;  
    } else {  
        return fib(n-1) + fib(n-2);  
    }  
}
```

$$1 + 0 = 1$$

~~fib(1), n=1~~ | ~~fib(0), n=0~~

$$1 + 0 = 1$$

~~fib(2), n=2~~ | ~~fib(1), n=1~~

~~fib(1), n=1~~ | ~~fib(0), n=0~~

$$1 + 0 = 1$$

~~fib(1), n=1~~ | ~~fib(0), n=0~~

~~fib(3), n=3~~ | ~~fib(2), n=2~~

~~fib(2), n=2~~ | ~~fib(1), n=1~~

~~fib(4), n=4~~ | ~~fib(3), n=3~~

~~fib(5), n=5~~

~~Main, n=5 ← Ejemplo~~

Valor Retornado = 5

Callstack Busqueda Binaria

```
public int busquedaBinaria(int arreglo [], int izq, int der, int numBuscado){  
    int posicion = (izq + der)/2;  
    if (izq > der){  
        return -1;  
    } else if (arreglo[posicion] == numBuscado){  
        return posicion;  
    } else if (arreglo[posicion] < numBuscado){  
        return busquedaBinaria(arreglo, (posicion+1), der, numBuscado);  
    } else {  
        return busquedaBinaria(arreglo, izq, (posicion-1), numBuscado);  
    }  
}
```

condición ($29 == 29$)

~~busquedaBin (arreglo, 3, 3, 29), izq=3, der=3, posicion=3~~

condición ($8 < 29$)

~~busquedaBin (arreglo, 2, 3, 29), izq=2, der=3, posicion=2~~

condición ($5 < 29$)

~~busquedaBin (arreglo, 0, 3, 29), izq=0, der=3, posicion=1~~

condición ($34 > 29$)

~~busquedaBin (arreglo, 0, 9, 29), izq=0, der=9, posicion=4~~

~~Main, arreglo: [4, 5, 8, 29, 34, 41, 70, 83, 98, 115], x=29~~

numBuscado

Valor Retornado

3