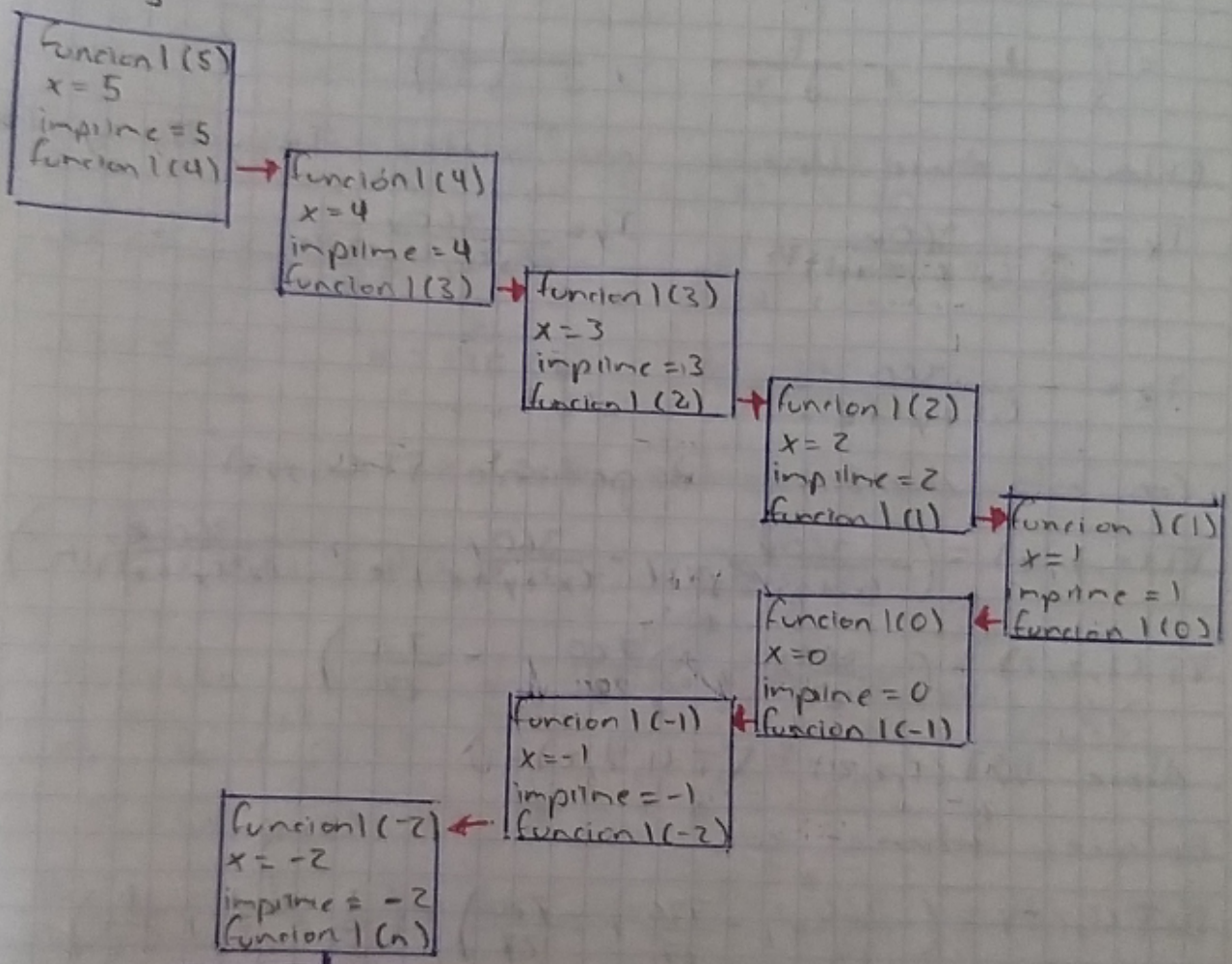
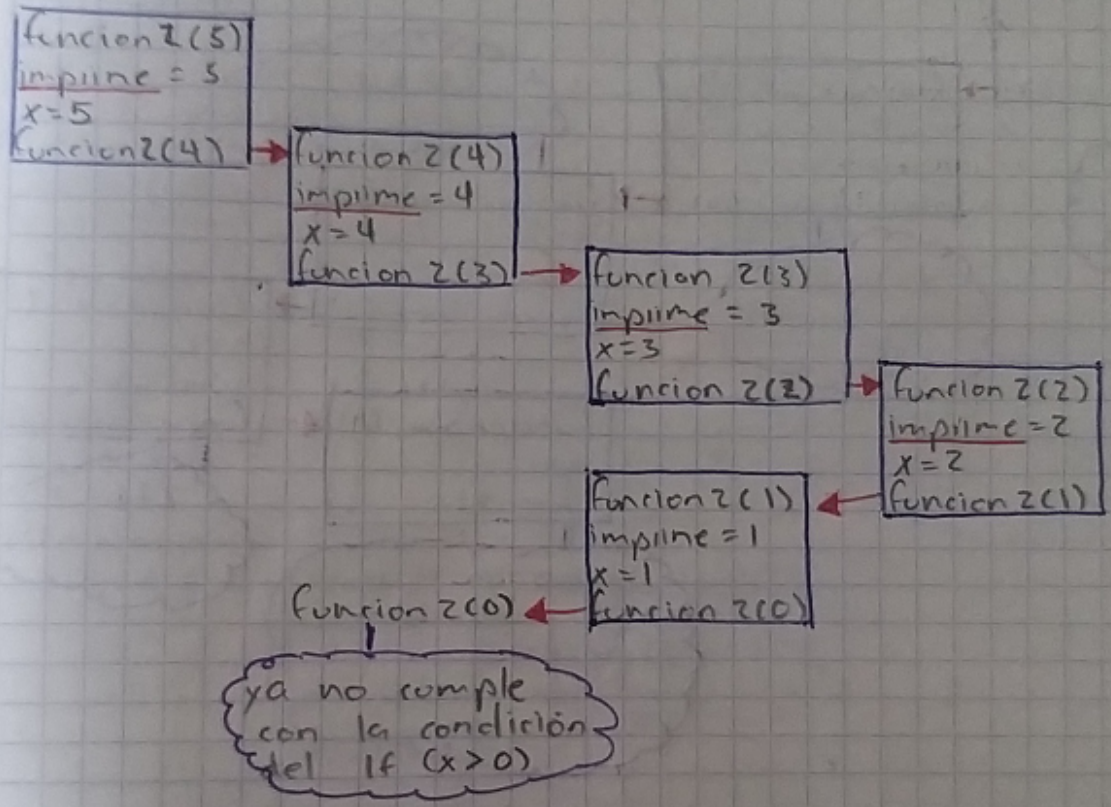
Imagen

```
public void funcion1 (int x) {
    System.out.println (x);
    funcion1 (x-1);
    }
public static void main (String [] args) {
    funcion1 (5);
    }
```



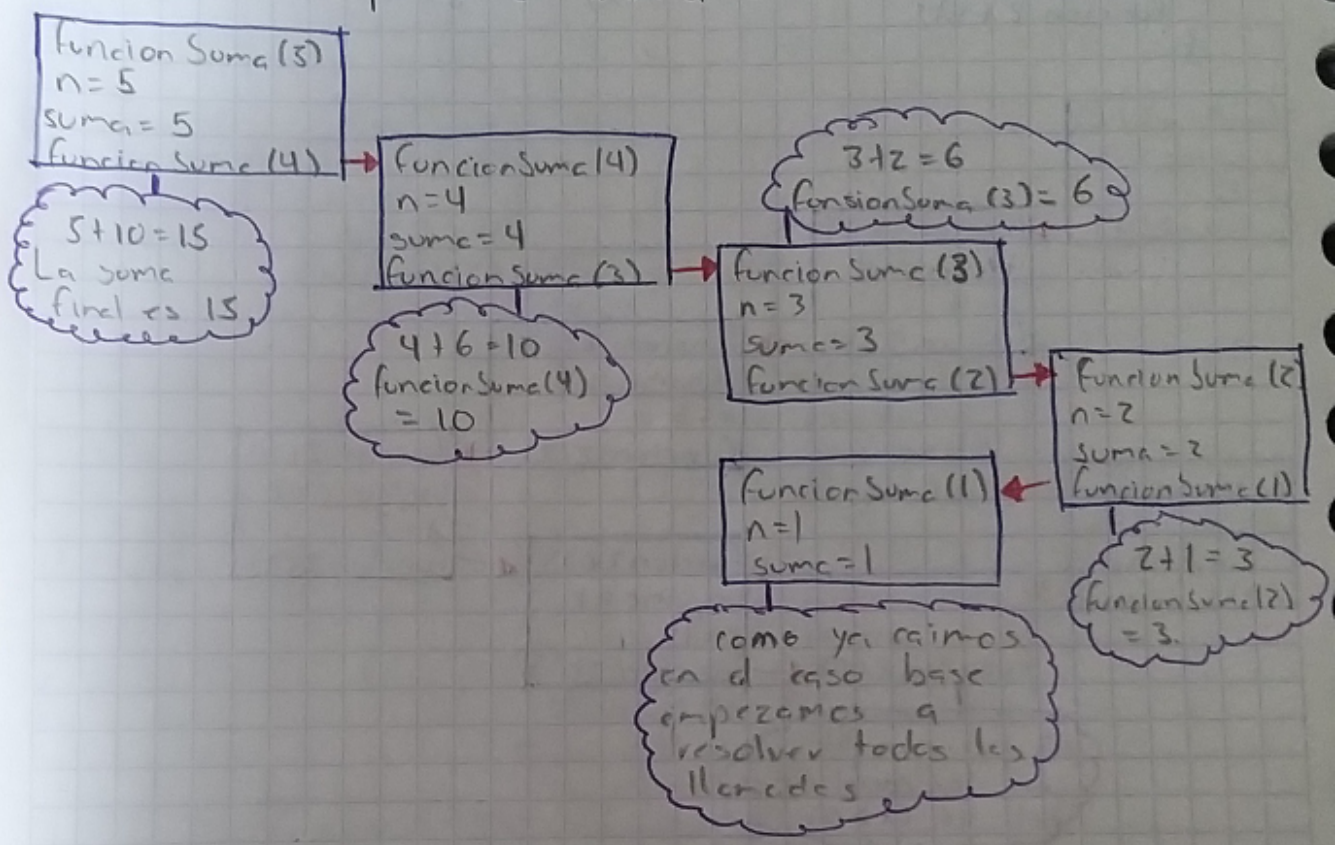
Este ejercicio solo va a correr hasta que la pila este llena ya que no tiene ningun caso base declarado.

```
• public static void funcion2 (int x) {
    if (x>0) {
        System.out.println (x);
        funcion2(x-1);
    }
  }
  public static void main (String [] args) {
    funcion2 (5);
  }
```

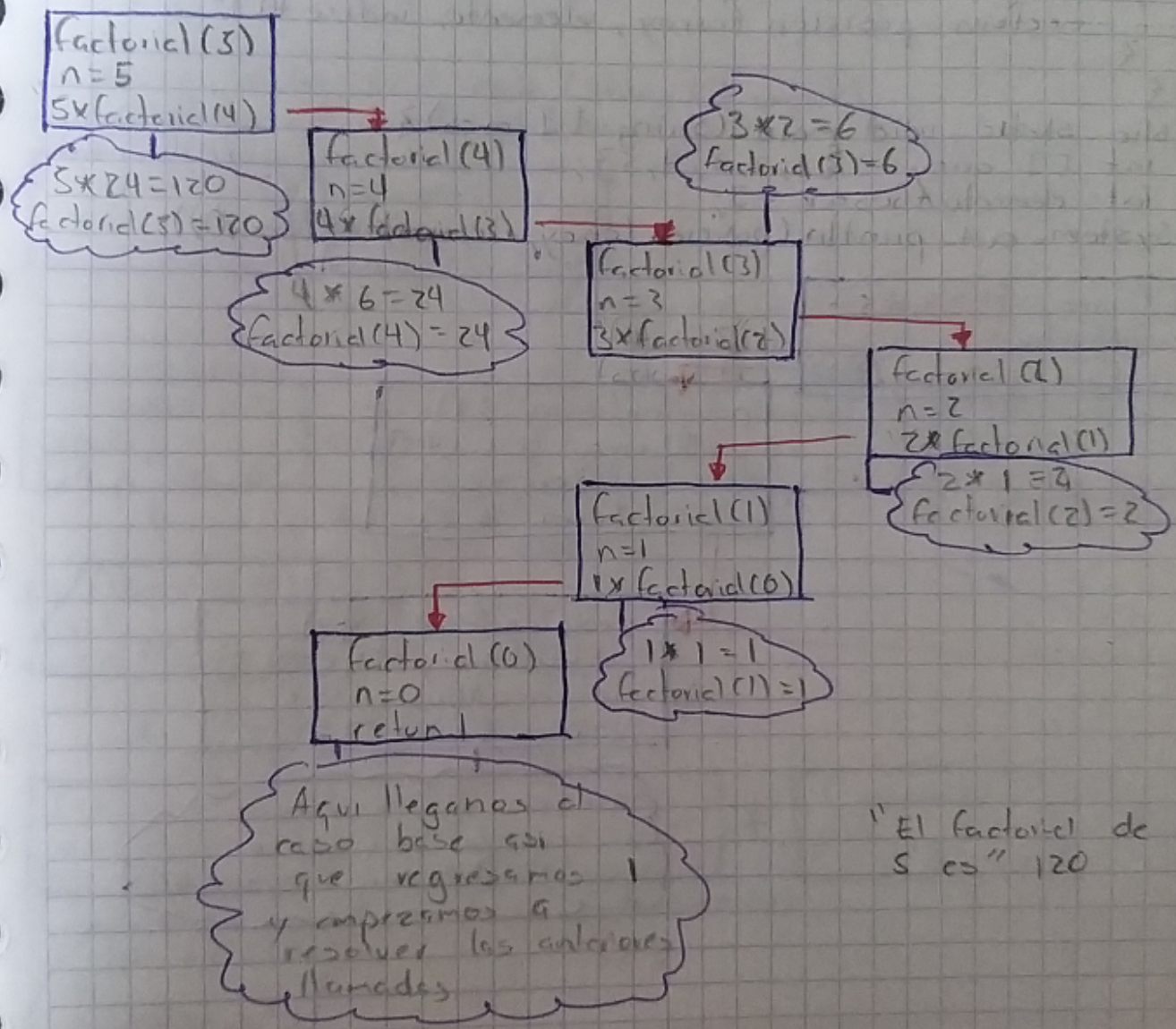


Se imprime
5
4
3
2
1

```
public static double funcionSuma (int n){
  If (n==1)
     return 1
  return n + funcionSuma (n-1);
}
public static void main (Strings [] args){
  System. out. println ("La suma es" + funcionSuma (5));
```



```
public static int factorial (int n){
  If ( n == 0)
    return 1
  return (n* factorial (n-1));
}
public static void main (String [] args) {
  System.out.println ("El factorial de 5 es" + factorial (5));
}
```



"El factorial de 5 es" 120

```
• public static int posicion(int[] array, int elemento, int indice){
  If(indice == array.length || array[indice] == elemento)
    If ( indice == array.length ){
      return -1;
    }else {
        return indice;
    }
  }else {
        return posicion (array, elemento, indice+1);
  }
}
public static void main (String [] args){
  int [] array = {1, 2, 3, 4, 5};
  int elementoABuscar = 3;
  System.out.println(posicion(array, elementoABuscar, 0));
```

```
int[] array = {1,2,3,4,5}
elementoABuscar = 3
S.O.P. (posicion(array,3,1))
```

```
posicion(array,3,1)
int[] array = {1,2,3,4,5}
elementoABuscar = 3
posicion(array, 3, 2)
```

```
posicion (array, 3,2)
int[] array = {1,2,3,4,5}
elementoABuscar = 3
posicion (array 3,3)
```

cae en el caso base y regresa el indice que andabamos buscando

```
public static int fibonacci (int n) {
  System.out.println ("fib de" n )
  if (n==0){
    return 0;
  } else if (n==1){
    return 1;
  } else {
    return fibonnaci (n-1) + fibonacci (n-2);
  }
}
public static void main (String [] args) {
  System.out.println (fibonacci (5));
```

fibonacci (1)
n=1
S.o.p. "fib de" 1

+

fibonacci (0)
n=0
S.o.p "fib de" 0

1 + 0 = 1 = fibonacci (2)

Luego vamos con la llamada de

fibonacci (2)
n=2
S.o.p "fib de" 2

+

fibonacci (1)
n=1
S.o.p "fib de" 1

+

fibonacci (1)
n=1
S.o.p "fibo de" 1

+

fibonacci (0)
n=0
S.o.p "fibo de" 0

1 + 1 = 2 = fibonacci (3) 1 + 0 = 1 = fibonacci (2)

+

fibonacci (1)
n=1
S.o.p "fibo de" 1

+

fibonacci (0)
n=0
S.o.p "fib de" 0

1 + 0 = 1 = fibonacci (2)

fibonacci (3)
n=3
S.o.p "fibo de" 3

+

fibonacci (2)
n=2
S.o.p "fibo de" 2

+

fibonacci (2)
n=2
S.o.p "fibo de" 2

+

2 + 1 = 3 = fibonacci (4) 1 +

fibonacci (1)
n=1
S.o.p "fibo de" 1

1 = 2 = fibonacci (3)

Fibonacci (4) n=4 S.o.p. "fib de" 4	+	Fibonacci (3) n=3 S.o.p "fibo de" 3
3	+	2 = 5 → fibonacci(5)

Ahora llegamos al resultado

Entonces

Fibonacci (5) n=5 S.o.p "fib de" 5

La forma recursiva trabaja desde fibonacci de 4 y luego va siguiendo hasta llegar a los casos base de fibo de 0 y 1 y después de ahí empieza a resolver los demás fibo hasta poder llegar a fibo de 5.

Imprime

fibo de 5
fibo de 4
fibo de 3
fibo de 2
fibo de 1
fibo de 0
fibo de 1
fibo de 2
fibo de 1
fibo de 0
fibo de 3
fibo de 2
fibo de 1
fibo de 0
fibo de 1