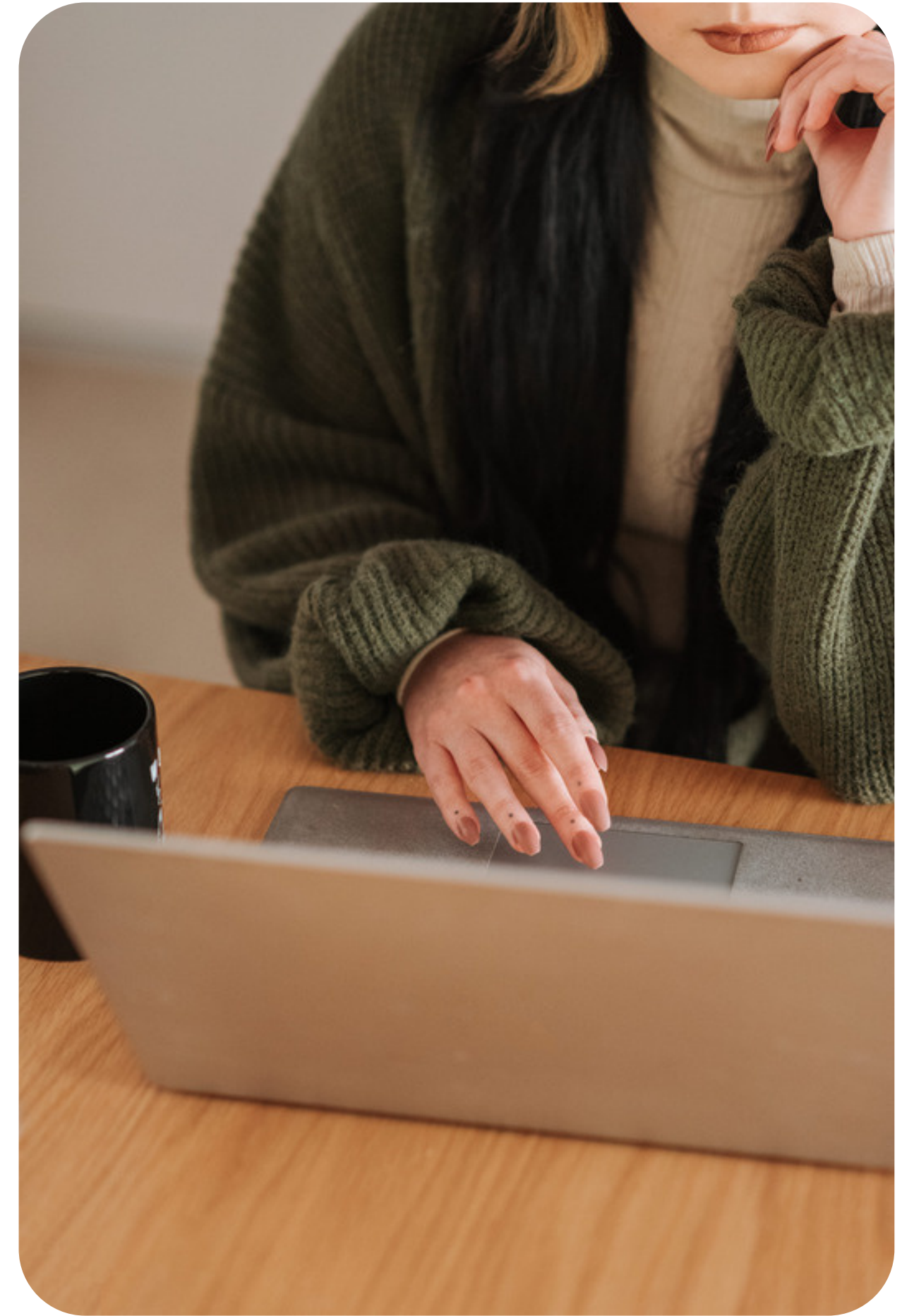


Clase 2

02/02/2022



Agenda



01

Lectura de la practica

02

Funciones

03

Recursividad

04

Dudas

1

Lectura de la practica

2

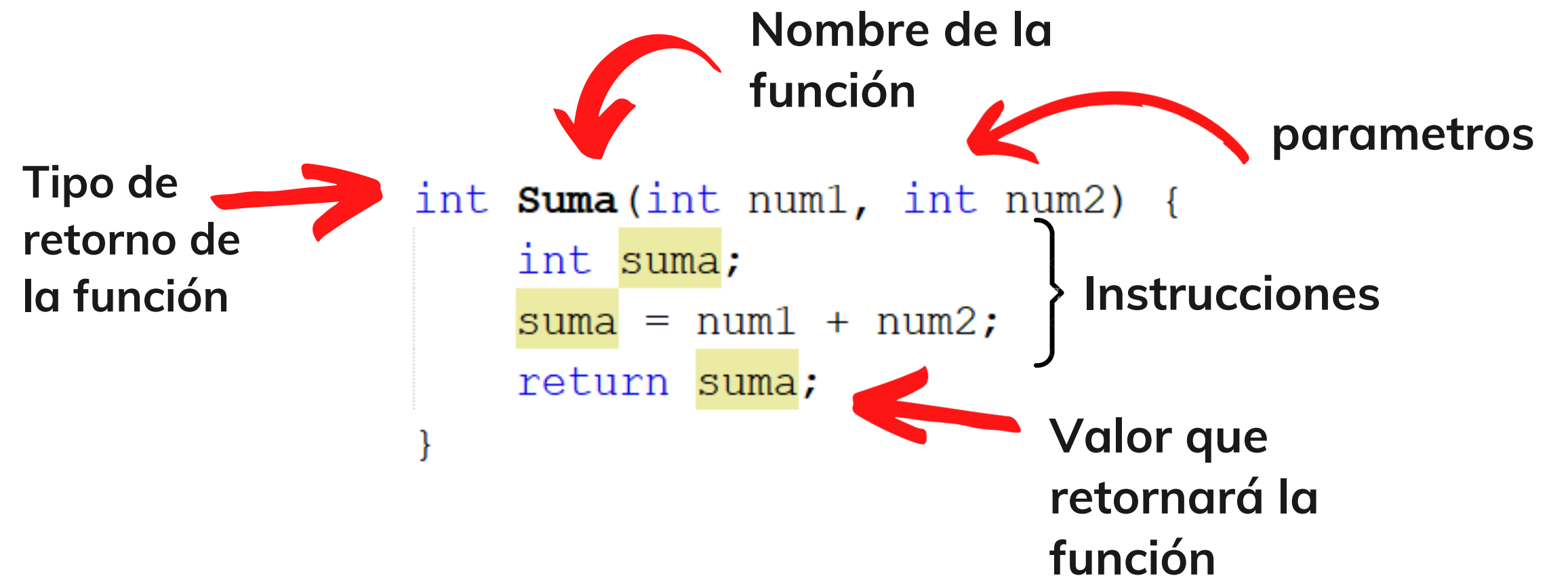
Funciones



¿QUÉ ES UNA FUNCIÓN?

- Es una secuencia de sentencias o instrucciones encapsuladas o aisladas

ESQUEMA DE UNA FUNCIÓN



Funciones

```
(define divide-list
  (lambda (l)
    (let divide
      ((l)
       (p '())
       (d '()))
      (if (null? l)
          (cons '() '())
          (cons (car l) (divide (cdr l) d (cons (car l) p)))))))

(define merge
  (lambda (s1 s2 . pred?)
    (let ((p (if (null? pred?) <= (car pred?))))
      (let merge ((s1 s1) (s2 s2))
        (cond ((null? s1) s2)
              ((null? s2) s1)
              ((p (car s1) (car s2)) (cons (car s1) (merge (cdr s1) s2)))
              (else (cons (car s2) (merge s1 (cdr s2)))))))

(define merge-sort
  (lambda (l . pred?)
    (let ((p (if (null? pred?) <= (car pred?))))
      (if (or (null? l) (null? (cdr l)))
          l
          (let ((divided-list (divide-list l)))
            (merge (merge-sort (car divided-list) p)
                    (merge-sort (cdr divided-list) p)))))))
```

CARACTERISTICAS DE UNA FUNCIÓN

- Es independiente al resto del programa
- Puede recibir parámetros.
- Retornan un valor
- Posee operaciones internas

LLAMADA

- Es invocar a un procedimiento o una función, proporcionando un valor útil para cada uno de los parámetros de entrada definidos, para ejecutar su lista de instrucciones.

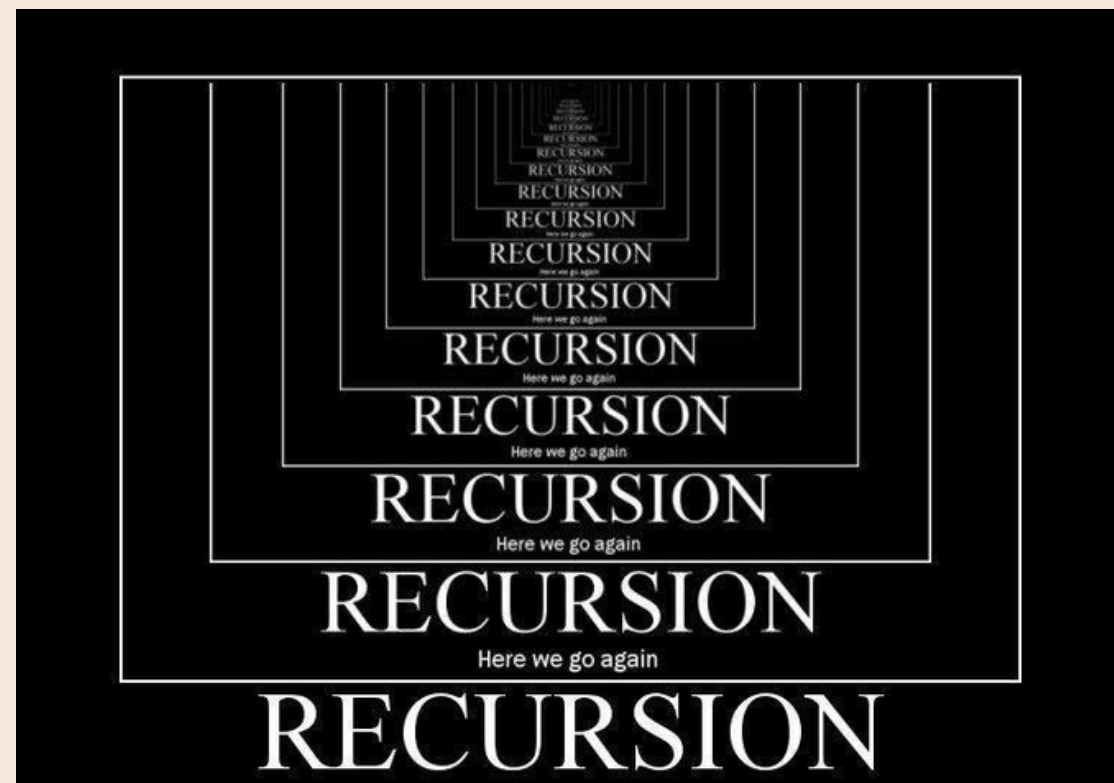
Llamada

```
public static void main(String[] args) {
    int resultado = Suma(1,2);
    System.out.print(resultado);
}

static int Suma(int num1, int num2) {
    int suma;
    suma = num1 + num2;
    return suma;
}
```


2

Recursividad



¿QUÉ ES RECURSIVIDAD?

- Es una función o procedimiento que dentro de su lista de instrucciones posee al menos una llamada a sí mismo.

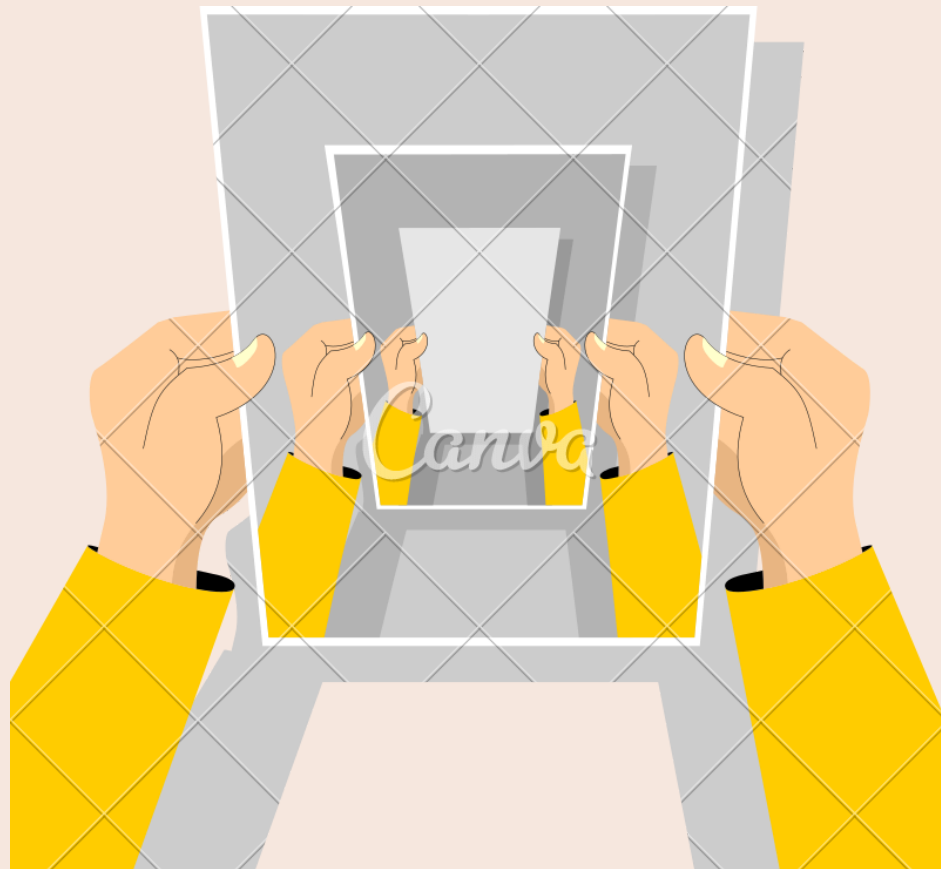
EJEMPLO

```
private void MiProcedimiento ( ){  
    int numero1 = 5;  
    int resultado = factorial( numero1 );  
    System.out.println( "El resultado del factorial es: " + resultado );  
}
```

```
public int factorial( int numero) {  
    if ( numero > 1 ){  
        int resultado= numero * factorial( numero - 1 );  
    }else {  
        resultado = 1;  
    }  
    return resultado;  
}
```

2

Recursividad



RECURSIVIDAD DOBLE Y SIMPLE

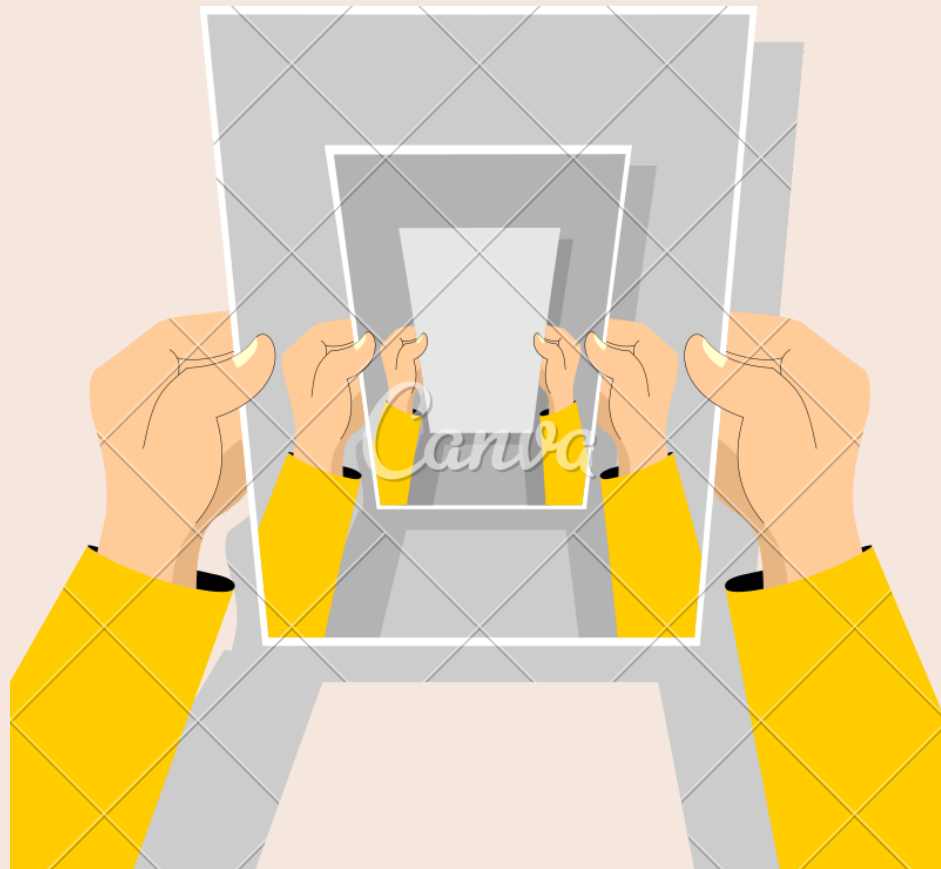
- Hay dos tipos de recursividad según la cantidad de veces que se llama así misma la función dentro de las instrucciones

EJEMPLO RECURSIVIDAD SIMPLE

```
public int factorial( int numero) {  
    if ( numero > 1 ){  
        int resultado= numero * factorial( numero - 1 );  
    }else {  
        resultado = 1;  
    }  
    return resultado;  
}
```

2

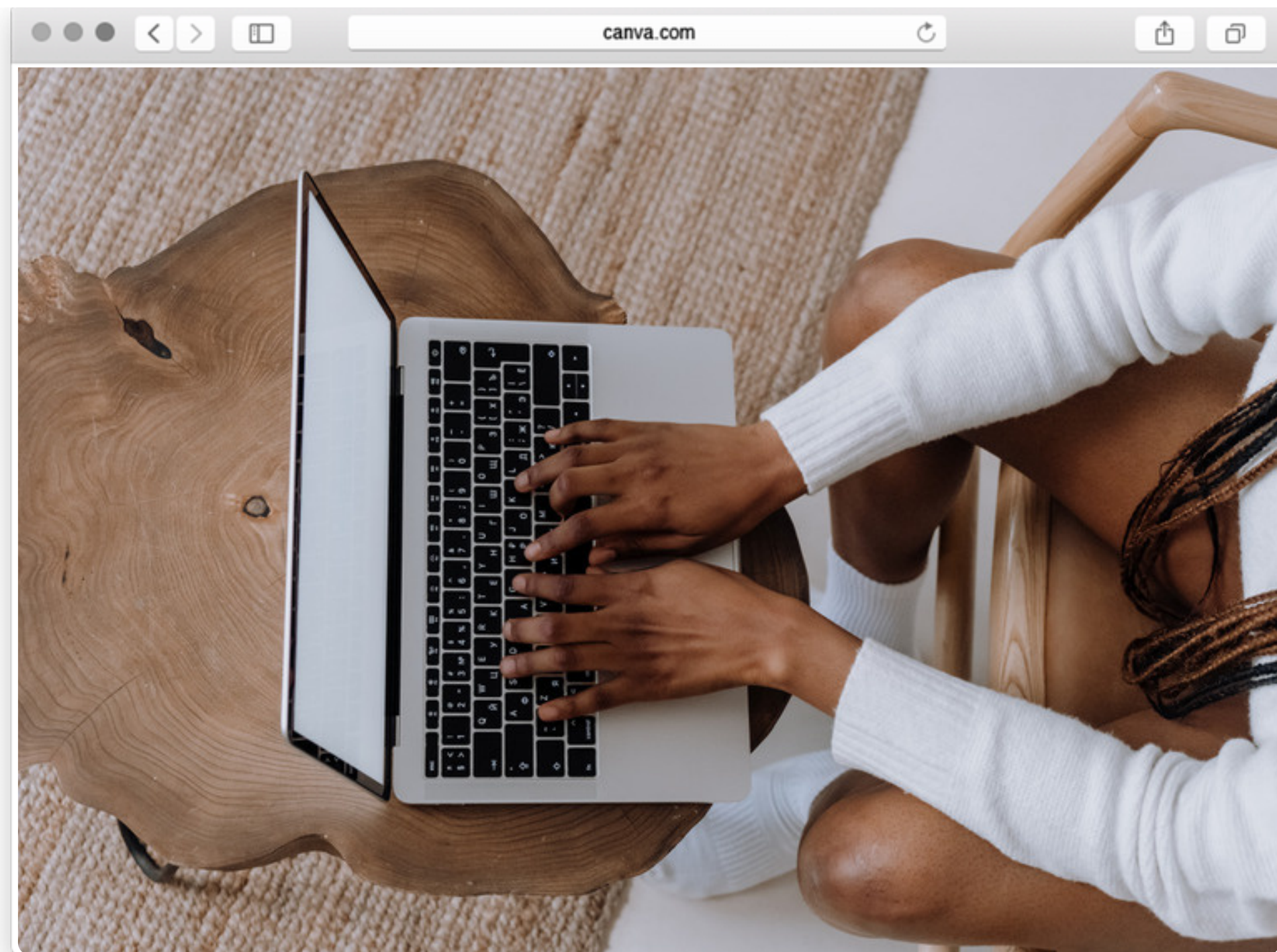
Recursividad



EJEMPLO RECURSIVIDAD DOBLE

```
public int Fib( int n ) {  
    if(n<=1) {  
        return(1);  
    }  
    return(Fib(n-1) + Fib(n-2));  
}
```


¿Dudas?



Tarea 1



Ejemplo



¡Muchas gracias!

Tomen awa