CARRERA DE INGENIERÍA DE SISTEMAS

Guía de Programación: Fundamentos de Programación en Java

PROGRAMACIÓN ORIENTADO A OBJETOS

UNIVERSIDAD DE LIMA

TEMARIO

1. Estructura Condicionales en Java
2. Estructuras repetitivas

UNIVERSIDAD DE LIMA

2. ESTRUCTURAS CONDICIONALES EN JAVA

UNIVERSIDAD DE LIMA

2.3. Estructura if/else en Java

```
if (condición) {
    bloque-de-sentencias-if
}
else {
    bloque-de-sentencias-else
}
```

2.3. Estructura if/else en Java

- Instrucción if (simple)
  - Principal forma de llevar a cabo la toma de decisiones en los programas. Su forma básica es:

```
boolean referencia1 = true;
boolean referencia2 = false;

if( referencia1 == referencia2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}
```

operadores lógicos

```
int numero1 = 50;
int numero2 = 150;

if( numero1 <= numero2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}
```

operadores aritméticos

UNIVERSIDAD DE LIMA

2.3.1. Ejemplo sobre if/else en Java

```
public static void main(String[] args){
        // Dado dos números indicar cual es el mayor
        java.util.Scanner entrada = new java.util.Scanner(System.in);
        System.out.println("Ingrese el primer número: ");
        int a = entrada.nextInt();
        System.out.println("Ingrese el segundo número: ");
        int b = entrada.nextInt();
        if (a>b)
            System.out.println("El mayor es: "+a);
        else
            System.out.println("El mayor es: "+b);
    }
```

UNIVERSIDAD DE LIMA

Estructuras de Control de flujo: Estructuras Selectivas

- Instrucción if (doble)
  - Opcionalmente podemos utilizar la expresión if-else

```
boolean referencia1 = true;
boolean referencia2 = false;

if( referencia1 == referencia2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}

else
{
    //secuencia de instrucciones
    //caso contrario
}
```

operadores lógicos

```
int numero1 = 50;
int numero2 = 150;

if( numero1 <= numero2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}

else
{
    //secuencia de instrucciones
    //caso contrario
}
```

operadores aritméticos

UNIVERSIDAD DE LIMA

UNIVERSIDAD DE LIMA

2.4. Estructura anidada if/else en Java

```
if(condicion){
acción...
   if(condicion){
   acción...
   }else{
   acción...
   }
}else{
acción...
}
```

2.4. Estructura anidada if/else en Java

- Instrucción if (Múltiple)
  - Opcionalmente podemos anidar if-else o else if

```
int numero1 = 50;
int numero2 = 150;

if( numero1 <= numero2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}

else
{
    if(numero1>numero2)
    {
        //secuencia de instrucciones
        //en forma secuencial
    }

}
```

```
int numero1 = 50;
int numero2 = 150;

if( numero1 <= numero2)
{
    //secuencia de instrucciones
    //si la expresion es verdadera
}

else if(numero1>numero2)
{
    //secuencia si cumple
    //con la siguiente condicion
}

else if(numero1>=2)
{
    //secuencia si cumple
    //con la siguiente condicion

}
```

UNIVERSIDAD DE LIMA

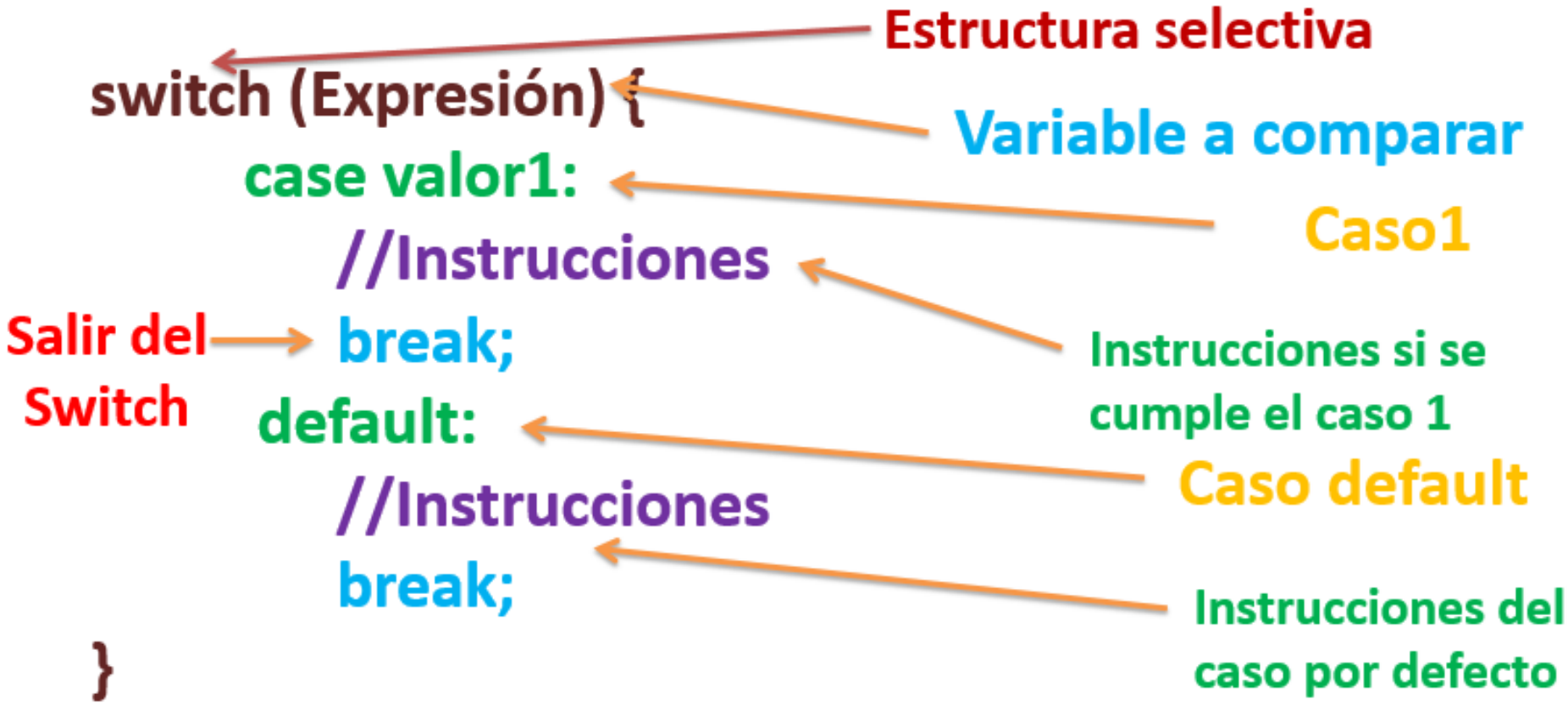
2.4.1. Ejemplo sobre if/else anidado en Java

```
public static void main(String[] args){
    // Dado el número de mes indicar su cantidad de días
    Scanner entrada = new Scanner(System.in);
    System.out.println("Ingrese el número de mes: ");  int mes = entrada.nextInt();
    if (mes==1 || mes==3 || mes==5 || mes==7 || mes==8 || mes==10 || mes==12){
        System.out.println("El mes tiene 31 días");
    } else if (mes==2){
        System.out.println("El mes tiene 28 o 29 días");
    } else if (mes==4 || mes==6 || mes==9 || mes==11){
        System.out.println("El mes tiene 30 días");
    }else { System.out.println("Número de mes invalido..."); }
 }
```

UNIVERSIDAD DE LIMA

UNIVERSIDAD DE LIMA

2.5. Estructura switch en Java



2.5. Estructura switch en Java

- Instrucción switch
  - Se utiliza para elegir entre valores pequeños de tipo entero. Su forma general es:

```
switch(edad)
{
    case 25:
        System.out.println("es la edad 25");
        break;
    case 30:
        System.out.println("no es la edad");
        break;
    default:
        System.out.println("caso por default");
        break;
}
```

2.5.1. Ejemplo sobre if/else anidado en Java

```
public static void main(String[] args){
    // Dado el número de mes indicar su cantidad de días
    Scanner entrada = new Scanner(System.in);
    System.out.println("Ingrese el número de mes: ");
    int mes = entrada.nextInt();
    switch (mes){
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            System.out.println("El mes tiene 31 días");
            break;
        case 2:
            System.out.println("El mes tiene 28 o 29 días");
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            System.out.println("El mes tiene 30 días");
            break;
        default:
            System.out.println("El número de mes no es válido...");
            break;
    }
}
```

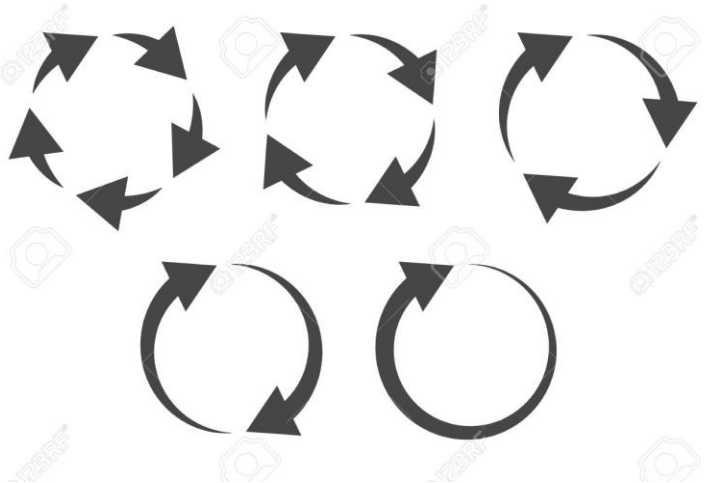
Operador condicional ?

- Operador condicional
  - En reemplazo de la sentencia if-else, podemos utilizar el operador condicional con los caracteres ? Y :
  - expresion1 ? expresion2 : expresion3
  - Si expresion1 es verdadera, entonces expresion2 es el valor condicional, en caso contrario expresion3 es el valor condicional

```
int bb=10;

System.out.println(bb%2==0?"par":"impar");
```

UNIVERSIDAD DE LIMA



3. ESTRUCTURAS REPETITIVAS EN JAVA

Estructura Repetitiva: Instrucción while

- Sintaxis:

```
While (condicion)
{       Instrucción 1;
        Instrucción 2;
        …
}
```

- Condición es una expresión booleana (puede ver verdadera o falsa) que se evalúa al principio del bucle y antes de cada iteración de las sentencias.
- Si condición es verdadera, se ejecuta el bloque de sentencias y se vuelve al principio del bucle.
- Si la condición es falsa, no se ejecuta el bloque de sentencias y se continúa con la siguiente sentencia del programa.

UNIVERSIDAD DE LIMA

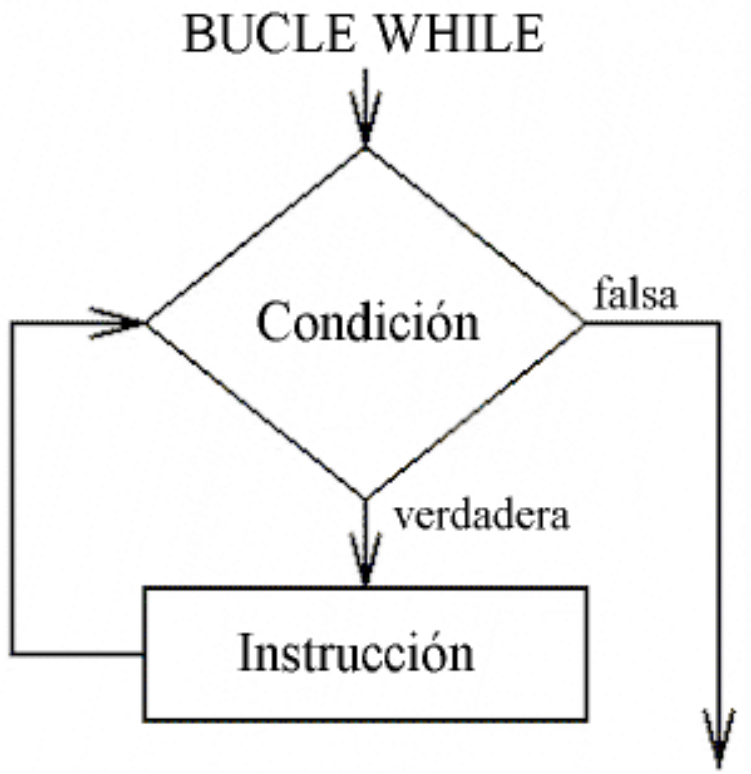
3.1. Estructura While en Java

Estructura Iterativa

Expresión Lógica

```
While (i<=n){
    s=s+1;
}
```

Instrucciones que se repetirán



3.1.1. Ejemplo sobre while en Java

```
public static void main(String[] args) {
    // Determinar el mayor de n números
    Scanner entrada = new Scanner(System.in);
    System.out.println("Ingrese n: ");    int n = entrada.nextInt();
    int i = 0; // Contador
    int mayor = 0;
    while (i < n) {
        i++;
        System.out.println("Ingrese un número: ");  int numero = entrada.nextInt();
        if (numero > mayor)  mayor = numero;
    }
    System.out.println("El mayor de n números es: "+mayor);
}
```

UNIVERSIDAD DE LIMA

Estructura Repetitiva: Instrucción for

- La instrucción for se ejecuta realizando primero la inicialización , después mientras que condición es verdadero se llevan a cabo las instrucciones y luego se realiza el incremento o decremento.
- Es posible anidar instrucción for
- Ejemplo:

```
for(int m=1;m<=20;m++)
{
    System.out.println(m);
}
```

UNIVERSIDAD DE LIMA

UNIVERSIDAD DE LIMA

3.2. Estructura for en Java

Estructura Iterativa

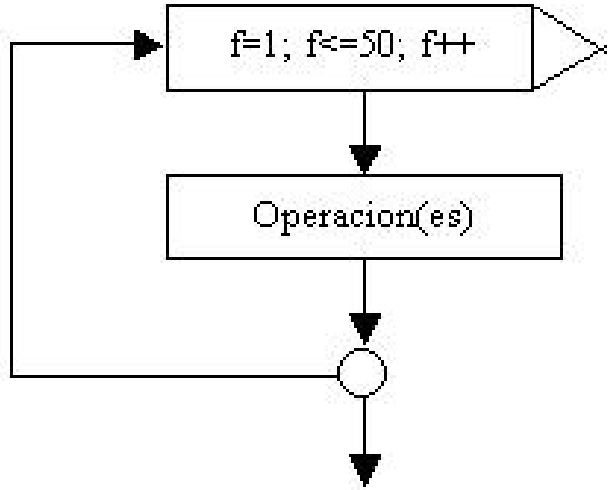
Inicialización

Expresión Booleana

Actualización

```
for (int i=1; i<=12; i++) {
    suma= num + i;
}
```

Instrucciones que se repetirán



3.2.1. Ejemplo sobre for en Java

```
public static void main(String[] args) {
        // Determinar el mayor de n números
        Scanner entrada = new Scanner(System.in);
        System.out.println("Ingrese n: ");  int n = entrada.nextInt();
        int mayor = 0;
        for (int i = 0; i < n; i++) {
            System.out.println("Ingrese un número: "); int numero = entrada.nextInt();
            if (numero > mayor) {
                mayor = numero;
            }
        }
        System.out.println("El mayor de n números es: " + mayor);
    }
```

UNIVERSIDAD DE LIMA

Estructura Repetitiva: Instrucción do

- Sintaxis:

```
do
{       Instrucción 1;
        Instrucción 2;
        …
}
 while (condición);
```

- El bloque de instrucciones se repite MIENTRAS que la condición sea verdadera.
- Es una estructura repetitiva que se ejecuta al menos una vez, a diferencia de las anteriores estructuras.
- La comprobación se ejecuta después de haber realizado el grupo de instrucciones especificadas.

UNIVERSIDAD DE LIMA

UNIVERSIDAD DE LIMA

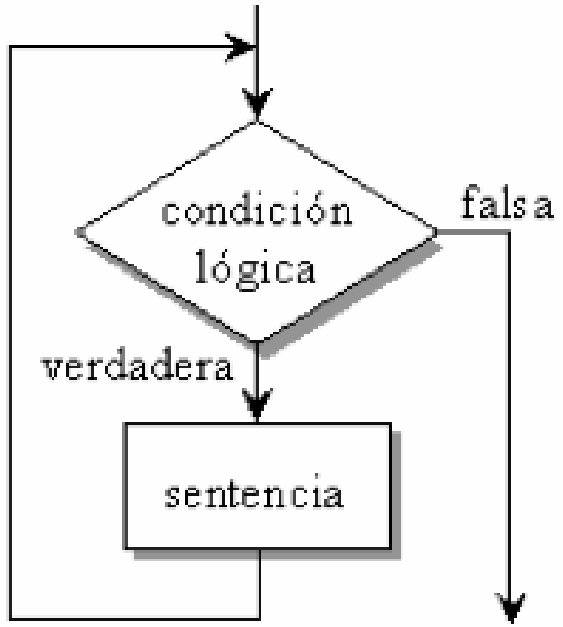
3.3. Estructura do-while en Java

Estructura Iterativa

```
do{
    s=s+1;
}
While (nota<=0 || nota>=20)
```

Instrucciones que se repetirán

Expresión Lógica



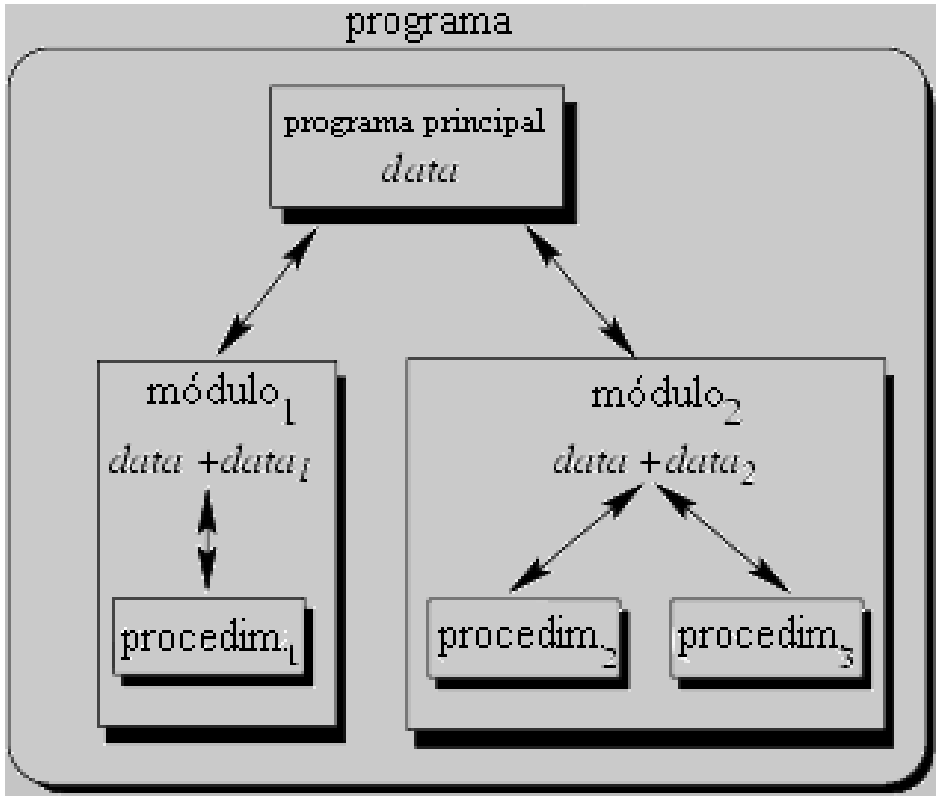
3.3.1. Ejemplo sobre do-while en Java

```
public static void main(String[] args) {
        // Determinar el mayor de n números
        Scanner entrada = new Scanner(System.in);
        System.out.println("Ingrese n: ");  int n = entrada.nextInt();
        int mayor = 0;
        int i = 0;
        do {
            i++;
            System.out.println("Ingrese un número: ");  int numero = entrada.nextInt();
            if (numero > mayor) { mayor = numero; }
        } while (i<n);
        System.out.println("El mayor de n números es: " + mayor);
    }
```

UNIVERSIDAD DE LIMA

3.4. Comparando estructura de bucle

- Usa el bucle while para iterar indefinidamente sentencias de cero o más veces.
- Use el bucle do-while para iterar indefinidamente instrucciones de uno o más veces.
- Utilice el bucle for para repetir las instrucciones un número predefinido de veces.



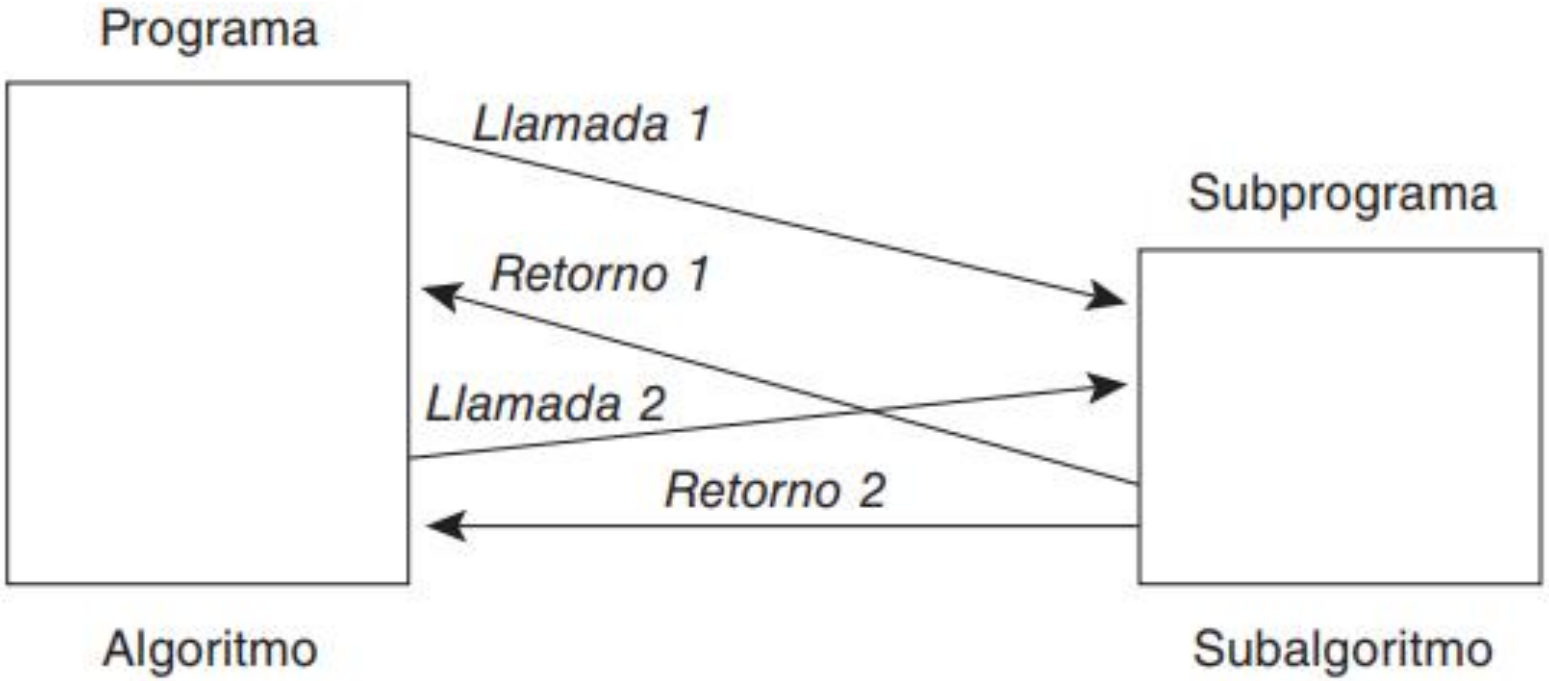
4. MODULARIDAD O SUBPROGRAMAS EN JAVA

Solución de un problema con modulos

- El problema principal se soluciona por el correspondiente programa o algoritmo principal —también denominado controlador o conductor (driver)— y la solución de los subproblemas mediante subprogramas, conocidos como procedimientos (subrutinas) o funciones. Los subprogramas, cuando se tratan en lenguaje algorítmico, se denominan también subalgoritmos.

UNIVERSIDAD DE LIMA

Creando e invocando métodos



Ejemplo de creación y llamado de un método

```
public class CalculatorTest {
    public static int sum(int numberOne, int numberTwo){
     return numberOne + numberTwo;
    }
   public static void main(String [] args) {
    int totalOne = sum(2,3);
    System.out.println(totalOne);
   }
}
```

UNIVERSIDAD DE LIMA