



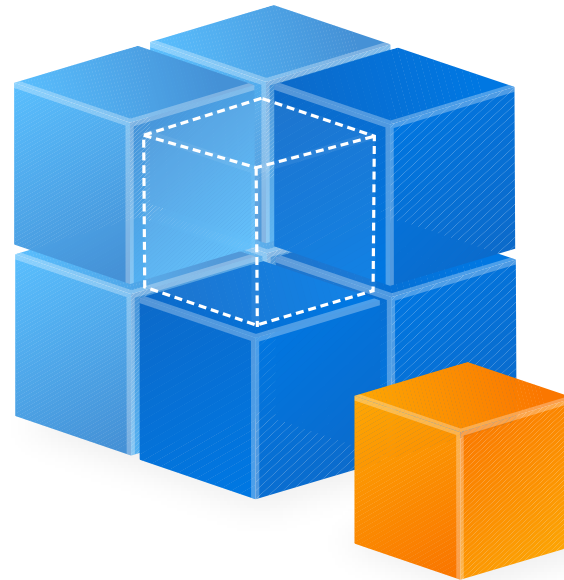
Funciones y Procedimientos

09/21/22

Introducción a la Programación – Pensamiento
Computacional

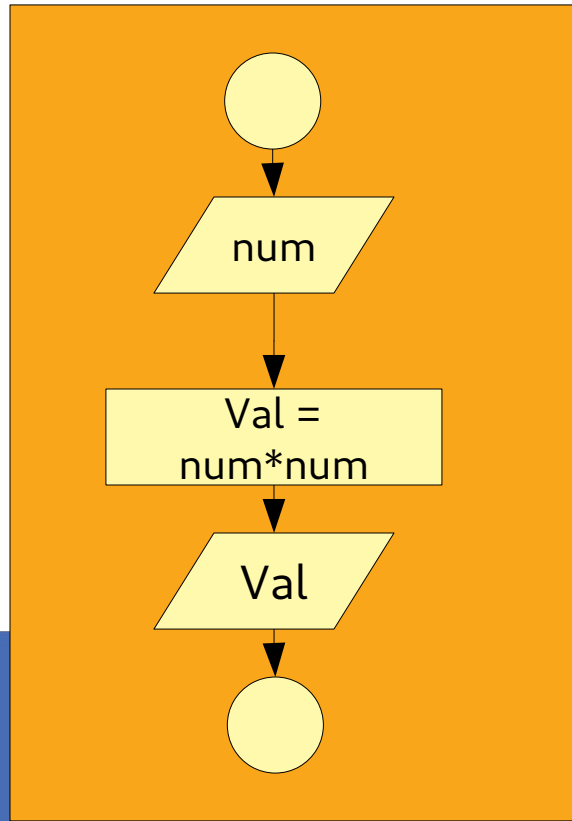
Subrutina

- Sub-programa
- Subalgoritmo que pertenece a un algoritmo principal, el cual permite resolver una tarea específica

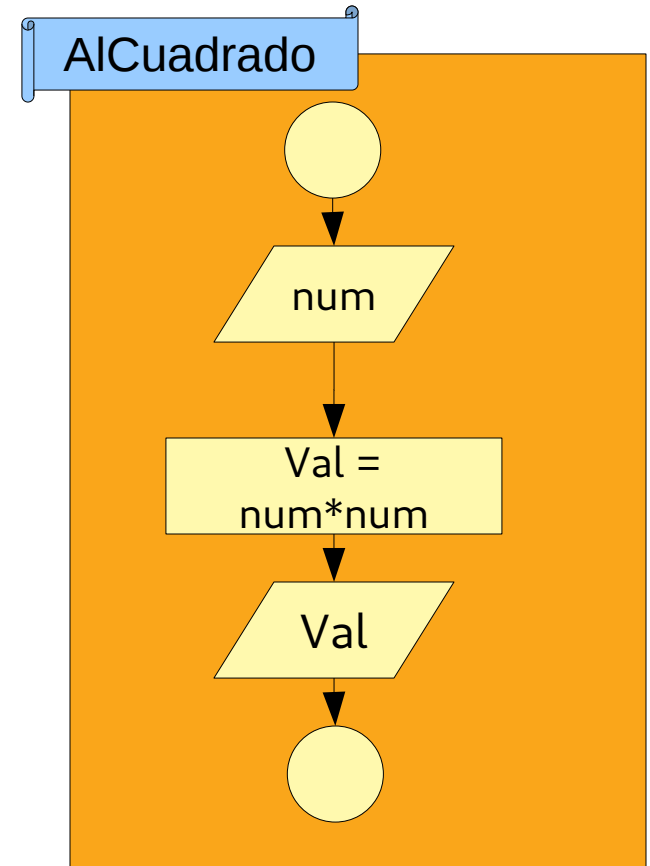
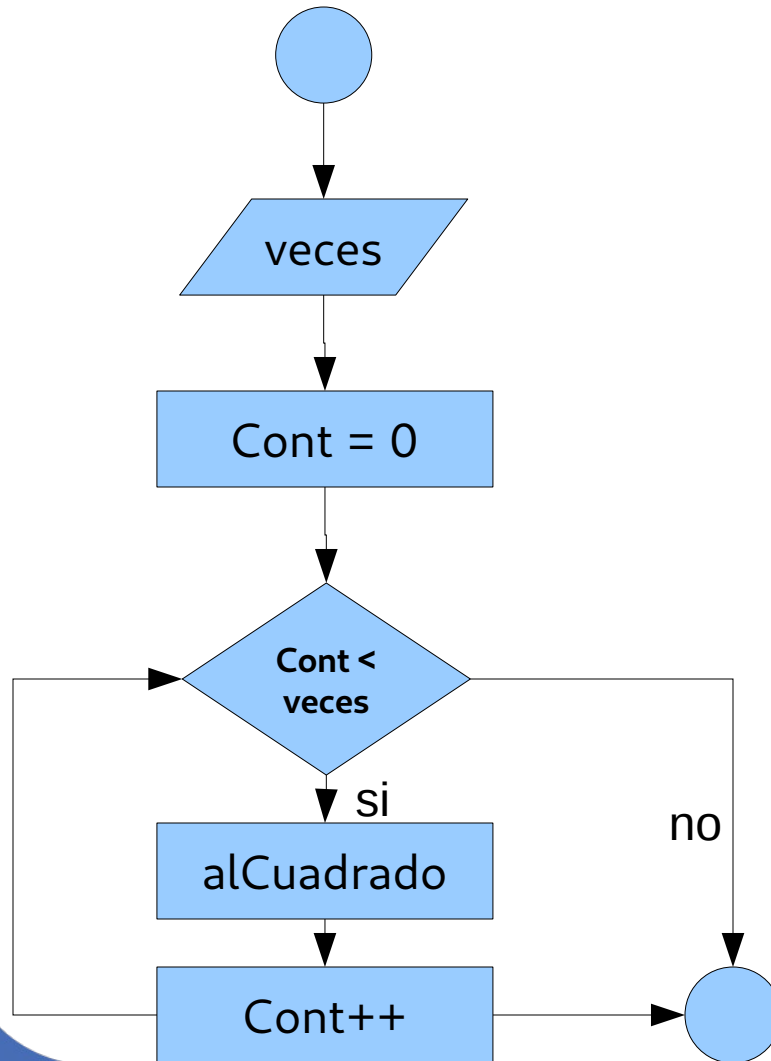


Subrutina

- Segmento de código separado del bloque principal, el cual puede ser invocado en cualquier momento desde ésta u otra subrutina.
- Una subrutina, al ser llamada dentro de un programa, hace que el código principal se detenga y se dirija a ejecutar el código de la subrutina.

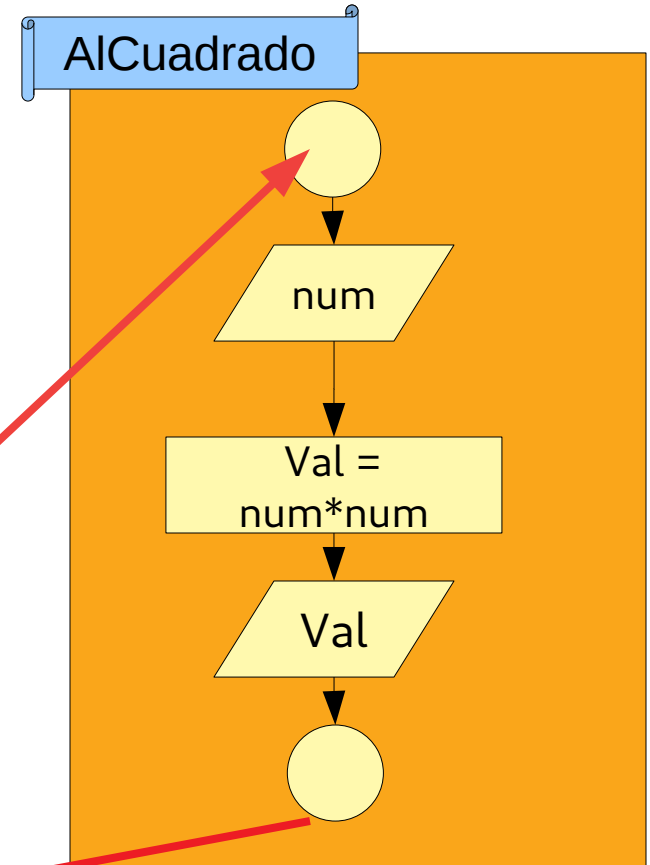
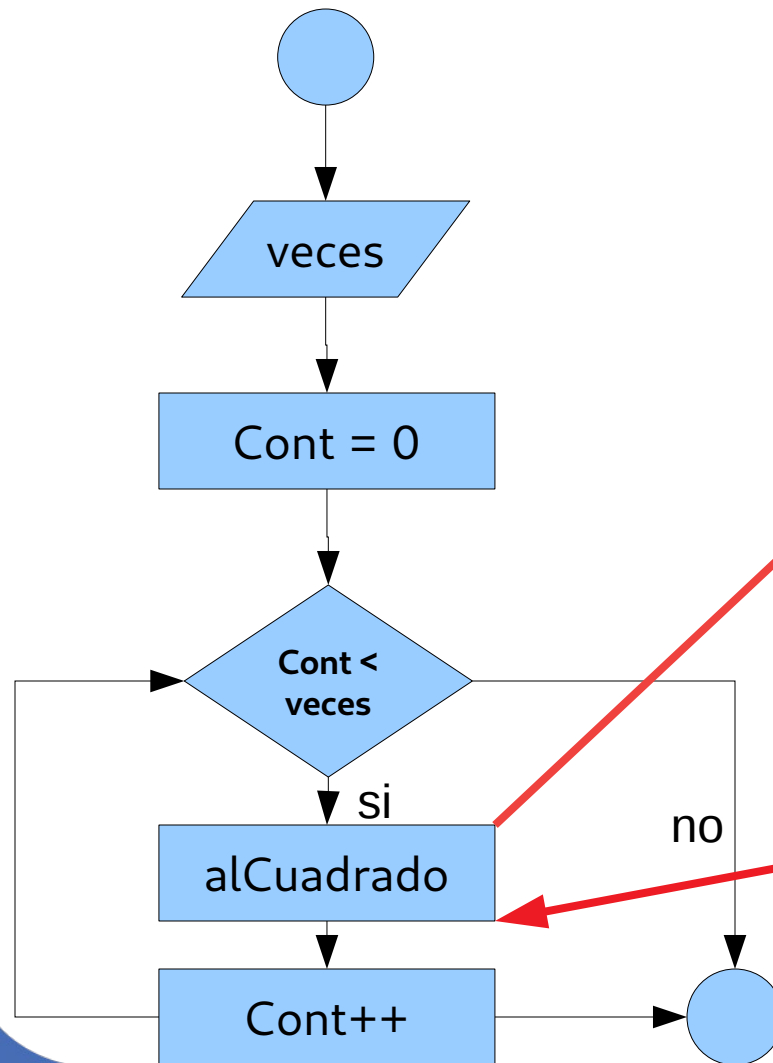


Subrutina



Algoritmo principal

Subrutina



Al terminar devuelve el control al algoritmo principal

... ya las hemos utilizado

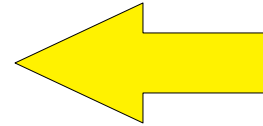
- `Console.WriteLine`

Tipos de Subrutinas

- Funciones
 - Son aquellas subrutinas que al terminar devuelven un valor
- Procedimientos
 - No devuelven un valor

Funciones

- Como en las calculadoras
 - pow
 - sqrt
- Componentes
 - Tipo de dato de retorno
 - Nombre (identificador)
 - Valor a devolver



Devuelven el valor
resultado de la
operación

Funciones

- Sintáxis declaración

```
tipo_resultado nombre_funcion (lista de parámetros)
{
    <acciones>
    RETURN (expresión)
}
```

Ejemplo:

```
public static int alCuadrado(int num){
    int res = num * num;
    return res;
}
```

La sentencia
RETURN,
retorna al
algoritmo
principal

Funciones

- Invocación asignando a variable

id_variable = nombreFuncion (parámetros)

- Invocación mostrando resultado

Console.WriteLine(nombreFuncion(parámetros))

- Ejemplo

```
static void Main() {  
    double val = alCuadrado(5);  
    Console.WriteLine(val);  
}
```

Procedimiento

- Es una subrutina que no devuelve un valor.
- De los que hemos utilizado
 - Read
 - Write
- **Definición en pseudocódigo:**

```
void nombre_proc (lista de parámetros)
{
    <acciones>
}
```

void indica que no devuelve un valor

Procedimientos

- Invocación

nombre_proc (lista de parámetros)

- Ejemplo

```
public static void mostrarPar(int num){  
    if (num % 2 == 0){  
        Console.WriteLine("Es par");  
    } else {  
        Console.WriteLine("No es par");  
    }  
}
```

```
static void Main() {  
    mostrarPar(3);  
}
```

Estructura de programa con funciones y procedimientos

Clase MiClase

BEGIN

Int x

double alCuadrado()

{

...

}

void mostrarPar()

{

...

}

MAIN

BEGIN

...

END

END

Ejemplo

- Escribe una función llamada `ConversionMoneda` que permita convertir una cantidad de dinero dada en quetzales a dólares y devuelva el valor de la conversión. Para luego invocarlo 5 veces desde el programa principal

Ejemplo

- Cree un procedimiento que reciba la edad y **muestre** si es mayor o menor de edad. Desde el programa principal ejecute dicho procedimiento 3 veces

Programa principal