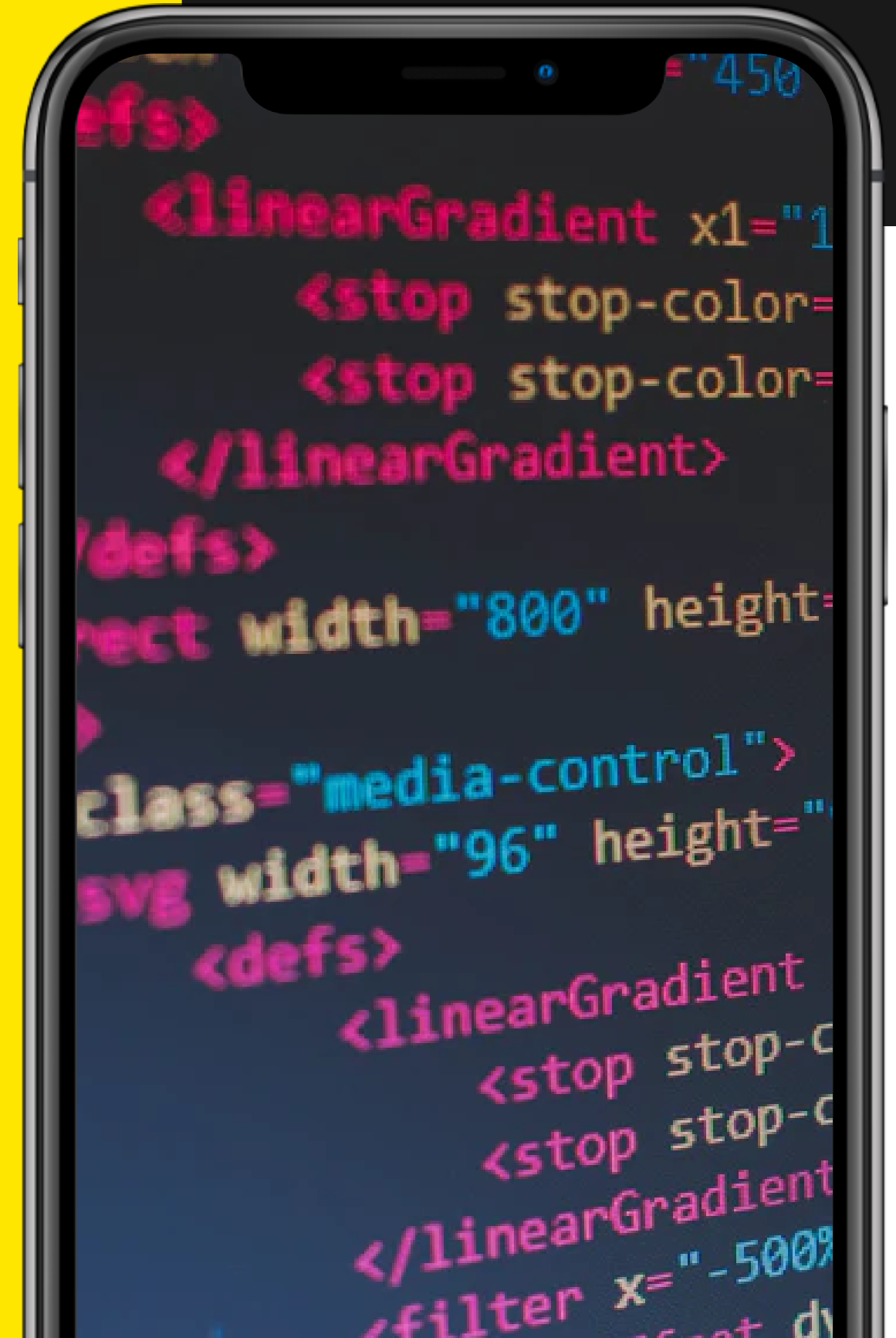
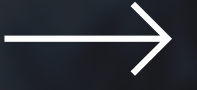


Clase 6



Contenido



Procedimiento/Metodo



Funciones

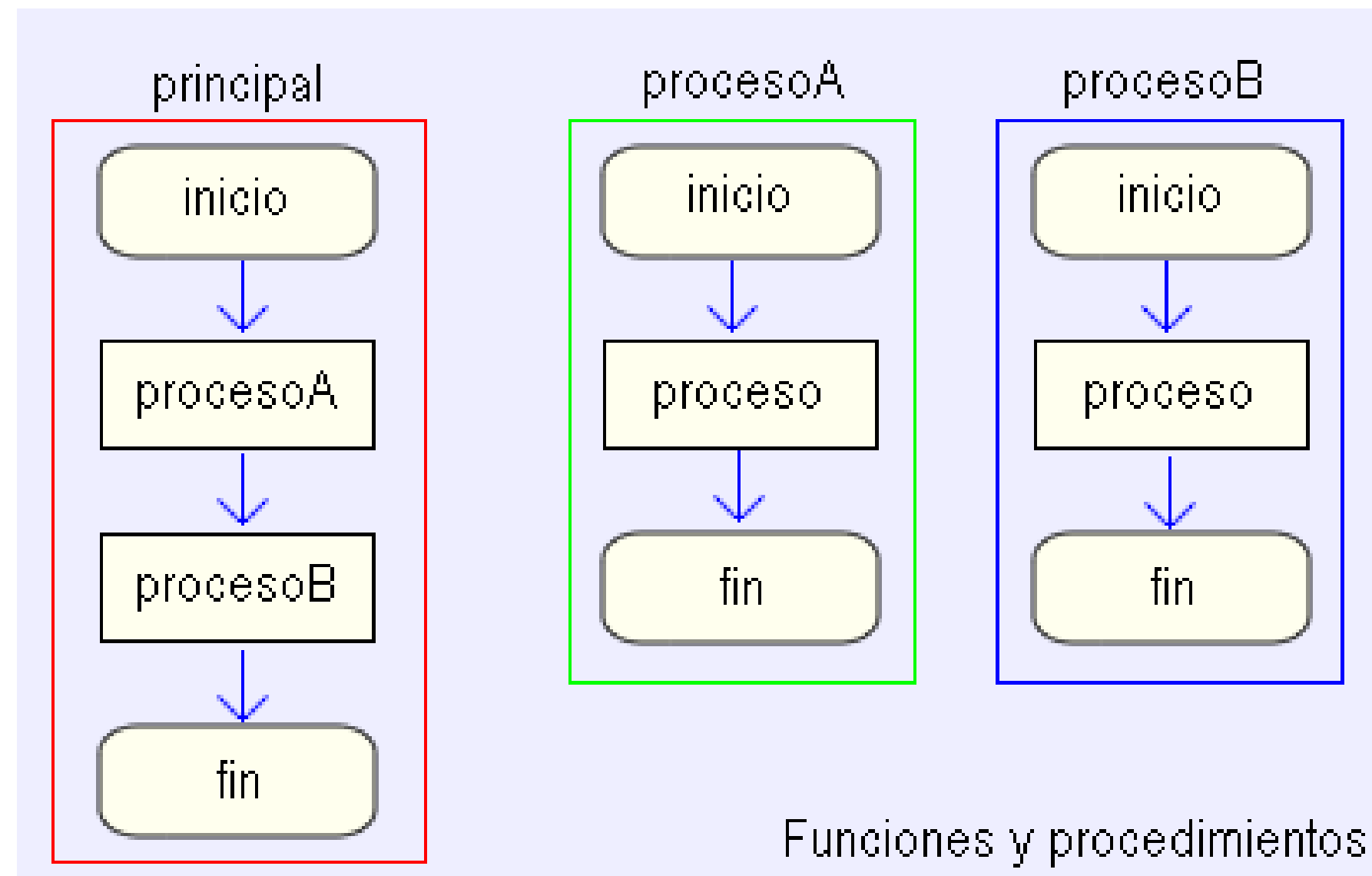


Parámetros

Procedimientos ó Metodos →

Son porciones de código que realizan una determinada tarea. Tienen un ambito que cumple con las mismas reglas que las variables.

Ejecutan un código a petición sin devolver ningún resultado.



Procedimientos ó Metodos →

Sintaxis

Ámbito de la declaración

Nombre del procedimiento

Instrucciones

```
private sub limpiar ()  
    txtNumero1.Clear()  
End sub
```

Procedimientos ó Metodos →

Sintaxis

La visibilidad de un procedimiento viene determinada por la declaración `private`, `public` o `internal`. Por defecto si no se indica nada se entiende que es `public`.

Diagram illustrating the syntax of a procedure with color-coded labels and arrows:

- Ámbito de la declaración** (Scope of the declaration) points to **private**.
- Nombre del procedimiento** (Name of the procedure) points to **sub limpiar ()**.
- Instrucciones** (Instructions) points to **txtNumero1.Clear()**.

```
private sub limpiar ()  
    txtNumero1.Clear()  
End sub
```

Procedimientos ó Metodos →

Sintaxis

Un procedimiento puede recibir o no parámetros, ejecuta instrucciones, pero nunca retorna ningún valor.

Diagram illustrating the syntax of a procedure with color-coded keywords and arrows pointing to their respective parts:

- Ámbito de la declaración** (Scope of declaration) points to **private**.
- Nombre del procedimiento** (Name of the procedure) points to **sub limpiar ()**.
- Instrucciones** (Instructions) points to **txtNumero1.Clear()**.

```
private sub limpiar ()  
    txtNumero1.Clear()  
End sub
```

Procedimientos ó Metodos →

Ejemplo

```
Sub limpiar()  
    Textbox1.clear()  
    Textbox2.clear()  
End Sub
```

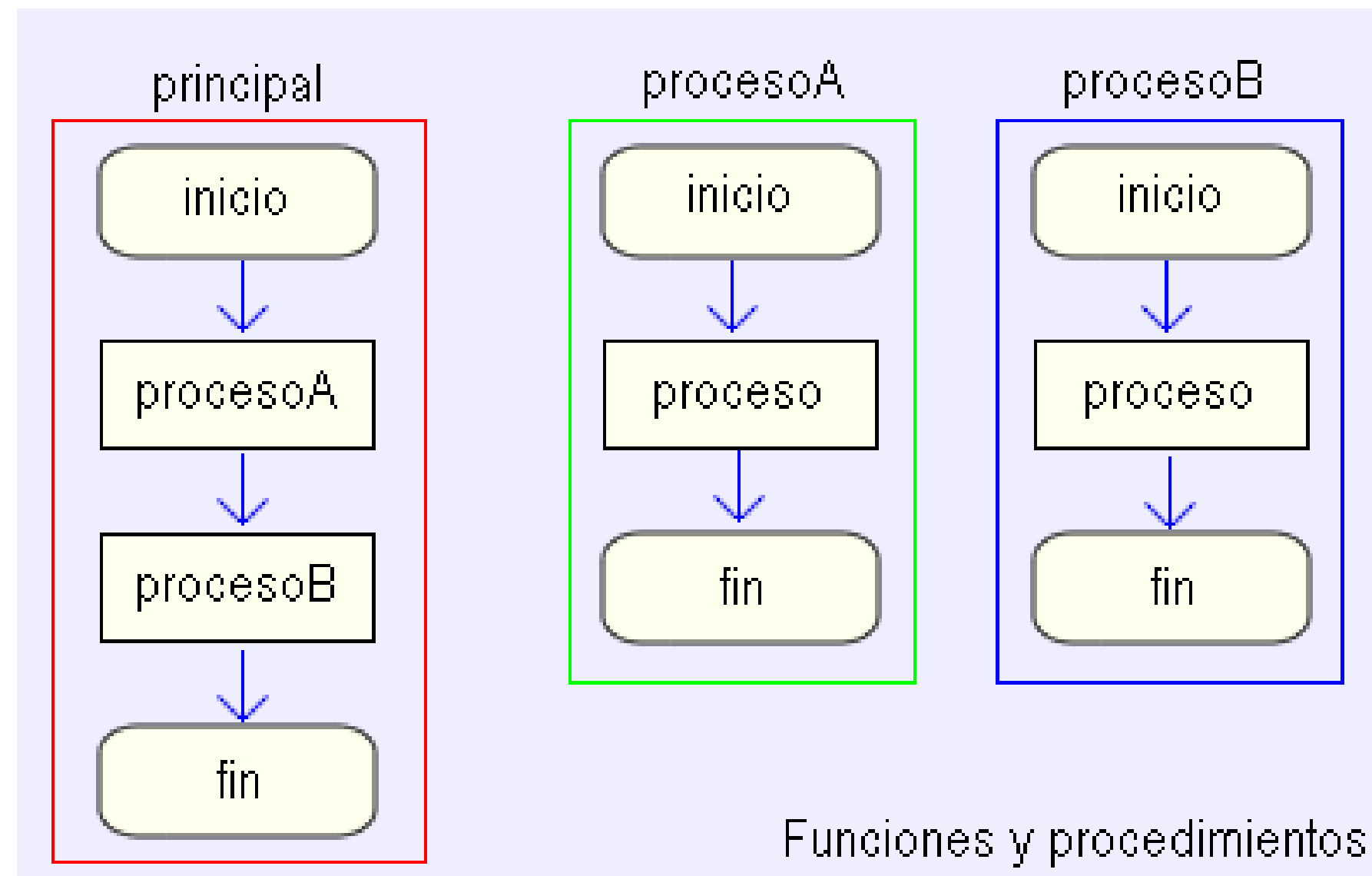


Funciones



Son porciones de código que realizan una determinada tarea. Tienen un ambito que cumple con las mismas reglas que las variables.

Ejecutan un código a petición y devuelven un resultado.



Funciones



Sintaxis

Ámbito de la función

Parámetros de entrada

Nombre de la función

Instrucciones

Valor de retorno

```
private Function sumar(numero1 as Integer,  
numero2 as Integer) as Integer  
    Dim suma as Integer = numero1 + numero2  
    Return suma  
End Function
```

Funciones



Pueden o no recibir parametros, pero a diferencia de los métodos, estos si retornan un valor.

The diagram illustrates the components of a VBA function with the following code and annotations:

```
private Function sumar(numero1 as Integer,  
numero2 as Integer) as Integer  
    Dim suma as Integer = numero1 + numero2  
    Return suma  
End Function
```

Annotations:

- Ámbito de la función** (green arrow) points to `private`.
- Parámetros de entrada** (red arrow) points to the parameter list `(numero1 as Integer, numero2 as Integer)`.
- Nombre de la función** (red arrow) points to the function name `sumar`.
- Instrucciones** (orange arrow) points to the body of the function, starting with `Dim suma as Integer = numero1 + numero2`.
- Valor de retorno** (orange arrow) points to the `Return suma` statement.

Comparacion



Ámbito de la declaración
Nombre del procedimiento
Instrucciones

```
private sub limpiar ()  
    txtNumero1.Clear()  
End sub
```

Ámbito de la función
Parámetros de entrada
Instrucciones
Nombre de la función
Valor de retorno

```
private Function sumar(numero1 as Integer,  
    numero2 as Integer) as Integer  
    Dim suma as Integer = numero1 + numero2  
    Return suma  
End Function
```

Funciones



Ejemplo

```
Function suma()  
    Dim numero1, numero2, resultado  
    numero1 = 5  
    numero2 = 3  
    resultado = numero1 + numero2  
  
    return resultado  
End Function
```

Métodos con Parámetros

Ejemplo 2



```
Sub ComputeArea(ByVal length As Double, ByVal width As Double)
    ' Declare local variable.
    Dim area As Double
    If length = 0 Or width = 0 Then
        ' If either argument = 0 then exit Sub immediately.
        Exit Sub
    End If
    ' Calculate area of rectangle.
    area = length * width
    ' Print area to Immediate window.
    Debug.WriteLine(area)
End Sub
```

Funcion con Parámetros

Ejemplo 2



```
Function Hypotenuse(side1 As Double, side2 As Double) As Double  
    Return Math.Sqrt((side1 ^ 2) + (side2 ^ 2))  
End Function
```

Llamada de la funcion:

```
Dim testLength, testHypotenuse As Double  
testHypotenuse = Hypotenuse(testLength, 10.7)
```