

# Computación II

## CIA-3351

### Unidad II

### Funciones

Carlos Valle Vidal  
Primer Semestre 2018

# Funciones

- Una función es un subprograma que permite resolver una tarea específica.
- Es una parte de un programa que usando su nombre puede ser invocada (ejecutada) desde otras partes tantas veces como se desee.
- Opcionalmente puede recibir valores (llamados argumentos o parámetros), se ejecuta y puede devolver un valor.

# Funciones predeterminadas

- Visual Basic nos provee de un conjunto de funciones (matemáticas, trigonométricas, de texto) que podemos utilizar directamente en nuestros programas.

# Funciones predeterminadas

- Funciones matemáticas y trigonométricas

Función	Descripción
ABS(x)	Devuelve el valor absoluto de x.
COS(x)	Devuelve el coseno de x
SIN(x)	Devuelve el seno de x
TAN(x)	Devuelve la tangente de x
SQR(x)	Devuelve la raíz cuadrada de x
LOG(x)	Devuelve el logaritmo natural de x
SGN(x)	Devuelve el signo de x (-1 si es negativo, 0 si es cero y 1 si es positivo)

# Funciones predeterminadas

- Funciones matemáticas y trigonométricas

Función	Descripción
ATN(x)	Devuelve el arcotangente de x.
ROUND(x,y)	Redondea x con y cifras decimales
RND(x)	Devuelve un número aleatorio

- Más funciones en:
- [https://www.techonthenet.com/excel/formulas/index\\_vba.php](https://www.techonthenet.com/excel/formulas/index_vba.php)

# Funciones predeterminadas

- Funciones que manipulan strings

Función	Descripción
INSTR(str1,str2)	Devuelve la posición de la primera ocurrencia de str2 en el string str1 (0 si no se encuentra el string)
Lcase(str1)	Devuelve el contenido del string str1 en minúsculas
Ucase(str1)	Devuelve el contenido del string str1 en mayúsculas
TRIM(str1)	Devuelve un string eliminando los espacios del principio y del final
REPLACE(str1,str2,str3)	Reemplaza la secuencia str2 por str3 dentro del str 1

# Funciones predeterminadas

- Funciones que manipulan strings

Función	Descripción
LEFT(str1,x)	Devuelve los primeros x caracteres de str1
RIGHT(str1,x)	Devuelve los ultimos x caracteres de str1
MID(str1,pos,x)	Devuelve el substring de largo x desde la posicion pos
len(str)	Devuelve el largo del string

# Construyendo Funciones

- Para declarar una función:

```
Function nombre_funcion (param1 As tipo_dato1,  
param2 As tipo_dato2) As tipo de dato
```

```
    Cuerpo de la función
```

```
    Nombre_función= valor de retorno
```

```
End Function
```



# Ejercicios

- Crear una función que calcule el factorial de un número
- Crear una función que retorne verdadero si el número es primo y falso si no lo es

# Fórmulas de Excel

- Excel tiene fórmulas que se pueden insertar en las planillas:
- [https://www.techonthenet.com/excel/formulas/index\\_ws.php](https://www.techonthenet.com/excel/formulas/index_ws.php)
- Para insertarla desde la macro Rango\_de\_celdas.Formula = Formula\_deseada

```
Sub calcular_promedios_mes()
```

```
    Dim i As Integer
```

```
    For i = 1 To 5
```

```
        Cells(6, i).Formula = "=Average(" & Cells(3, i).Address(0, 0) & ":"  
& Cells(5, i).Address(0, 0) & ")"
```

```
    Next i
```

```
End Sub
```

# Fórmulas de Excel

- Para insertar una formula en español:  
Rango\_de\_celdas.FormulaLocal = Formula\_deseada
- Información sobre referencias:

<https://exceltotal.com/referencias-en-excel/>

# Paso de parámetros por valor y por referencia

- En la mayoría de los lenguajes hay dos formas de pasar las variables a una función: **por valor** o **por referencia**.
- Por valor significa que la función (o subrutina) recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar.
- Por referencia significa que se pasa la posición de memoria donde esta guardada la variable, por lo que cualquier cambio de la función, modifica la variable que la llamó.
- En Visual Basic para excel, los parámetros se pasan por defecto por referencia.

# Paso de parámetros por valor y por referencia (2)

- Ejemplo:

```
Function funcion(a As Integer) As Integer
```

```
    a = a + 1
```

```
    funcion = a ^ 2
```

```
End Function
```

```
Sub macro6()
```

```
    Dim x As Integer
```

```
    x = 3
```

```
    MsgBox funcion(x)
```

```
    MsgBox x
```

```
End Sub
```

# Paso de parámetros por valor y por referencia (2)

- Muestra en pantalla 16 y luego 4, ya que la variable a se modifica, afectando también a la variable x de la macro que la llamó.
- Para explicitar que un parámetro será pasado por valor se debe poner la palabra **ByVal** antes del nombre de la variable.
- Ejemplo:

```
Function funcion(ByVal a As Integer) As Integer
```

```
    a = a + 1
```

```
    funcion = a ^ 2
```

```
End Function
```

# Paso de parámetros por valor y por referencia (2)

- De esta forma, nuestra macro anterior mostrará en pantalla 16 y luego 3, ya que el cambio en a no afecta el valor de x.
- Para explicitar que un parámetro será pasado por referencia se debe poner la palabra **ByRef** antes del nombre de la variable.

# Funciones con matrices como parámetros

- Para declarar una matriz como parámetro se hace de manera abierta:
- `Function nombre_funcion(nombrematriz() As tipo_dato) As tipo_dato`
- Esto obliga a que no puedan pasarse matrices por valor, solo pueden pasarse matrices por referencia.
- En general se recomienda pasar matrices y arreglos por referencia ya que se ahorra memoria.
- Ejemplo: `Function funcion(B() As Integer) As Integer`



# Funciones que devuelven matrices

- Para que una función devuelva arreglos o matrices, se debe indefinir la dimensión del arreglo o matriz en el retorno.
- Ejemplo: `Function nombre_funcion(nombre_variable As tipo_dato) As tipo_dato()`
- A diferencia de las funciones que hemos visto anteriormente, solo al momento de retornar el valor del arreglo o matriz se hace la asignación usando el nombre de la función.
- Es decir, dentro de la función debemos trabajar con una matriz o arreglo local que se vaya llenando. Una vez lleno, podemos devolverlo a la subrutina que la llamó usando el nombre de la función.

# Funciones que devuelven matrices (2)

- Ejemplo:

```
Function suma(x As Integer) As Integer()
```

```
    Dim i As Integer
```

```
    Dim a(5) As Integer
```

```
    for i = 0 to Ubound(a)
```

```
        a(i)= x*i
```

```
    Next i
```

```
    suma = a
```

```
End Function
```