

Carrera: **Analista en Sistemas Informáticos** (Modalidad: IRSO VIRTUAL)

Materia: **Programación II**

Profesor: **Daniel Lombardero**

**Clase N° 2 -  
práctica**

Fecha: 20/3/23

## **Proyectos simples**

Comenzamos con algo de práctica

**¡¡Hola a todos!!**

La intención de esta clase es que comencemos a tratar de practicar con algunas instrucciones sencillas de VB para ir acostumbrándonos a las formas.

Consigna 1: Generar una instrucción Select-Case para dos ejemplos que se les ocurran. ¿Qué diferencias hay entre el if y el Select-Case?

A)

Module Module1

```
Sub Main(args As String())  
    Dim name As String  
    name = "Guru99"  
    Select Case name  
        Case "John"  
            Console.WriteLine("Hello John")  
        Case "Guru99"  
            Console.WriteLine("Hello Guru99")  
        Case "Alice"  
            Console.WriteLine("Hello Alice")  
        Case "Joel"  
            Console.WriteLine("Hello Joel")  
        Case Else  
            Console.WriteLine("unknown name")  
    End Select  
    Console.WriteLine("VB.NET is easy!")  
    Console.ReadKey()  
End Sub  
End Module
```

B)

Module Module1

```
Sub Main(args As String())
```

```
    Dim Numero As Integer
```

```
    Numero = InputBox("Introduce un número del 1 al 5")
```

```
    Select Case Numero
```

```
        Case 1
```

```
            MsgBox "Has introducido el número 1"
```

```
        Case 2
```

```
            MsgBox "Has introducido el número 2"
```

```
        Case 3
```

```
            MsgBox "Has introducido el número 3"
```

```
        Case 4
```

```
            MsgBox "Has introducido el número 4"
```

```
        Case 5
```

```
            MsgBox "Has introducido el número 5"
```

```
    End Select
```

```
End Sub
```

Las **diferencias** son en el uso, si debes elegir entre varias opciones posibles dependiendo de una o varias condiciones, se usaría **CASE**, ya **que** permite juntar de manera más fácil las distintas opciones. Ahora si debes comprobar **que** una condición es verdadera o falsa y según esto asignar un valor, se usaría **IF**.

Consigna 2: Generar una instrucción Do-Loop para dos ejemplos que se les ocurran.

¿Qué diferencias hay entre el While y el Do-Loop?

A)

```
Sub Main(args As String())
```

```
    Dim i As Byte
```

```
    Dim s As Integer
```

```
    Dim fila As Byte
```

```
    Borra
```

```
    s = 0
```

```
    fila = 1
```

```
    Do
```

```
        i = Int(Rnd * 100) + 1
```

```
        s = s + i
```

```
        Cells(fila, 2) = i
```

```
        Cells(fila, 3) = s
```

```
fila = fila + 1
If s > 1000 Then End
Loop
End Sub
```

B)

```
Sub Main(args As String())

Dim factorial As Integer

Dim n As Integer

Dim dato As Integer

n = 1

dato = Range("A2") 'Se toma el numero del que se quiere obtener su factorial

'Factorial de cero es igual a uno por concepto

If dato = 0 Then

Factorial = 1

Else

Factorial = dato

Do While (n <> dato)

Factorial = Factorial * n

n = n + 1

Loop

End If

Range("B2") = Factoria

End Sub
```

Usando la estructura **while** sólo se pasa a ejecutar su contenido si se comprueba una condición lo **que** puede ocurrir 0, 1 o más veces. **Do Loop** funciona de forma similar, solo **que** nos aseguramos de **que** el contenido sea ejecutado al menos una vez, es decir **que** aunque su condición no se cumpla, su contenido se ejecuta.

Hasta aquí la clase práctica de hoy.

Nos mantenemos comunicados.

Saludos.

Daniel.