

Indicaciones: Captura los siguientes programas en Visual Basic 2005. Cada programa deberá estar contenido en proyectos por separado con el nombre de prog01, prog02, prog03, etcétera y los nombres de los módulos deberán de ser los que sugiere la línea de *Module* de cada programa.

Programa 1. Impresión de texto mediante la instrucción Write, Writeline y Read

```
Module welcome1
    Sub Main()
        Console.Write("Bienvenido a ")
        Console.WriteLine("Programacion orientados a objetos")
        Console.Read()
    End Sub
End Module
```

Programa 2. Comparación de dos números enteros utilizando las instrucciones Write, Read, Writeline y Readline y la estructura de control If Then Else

```
Module comparacion
    Sub Main()
        Dim n1, n2 As Integer
        Console.Write("Ingrese el primer número: ")
        n1 = Console.ReadLine()
        Console.Write("Ingrese el segundo número: ")
        n2 = Console.ReadLine()
        If (n1 = n2) Then
            Console.WriteLine("{0} es igual a {1}", n1, n2)
        End If
        If (n1 > n2) Then
            Console.WriteLine("{0} es mayor que {1}", n1, n2)
        End If
        If (n1 < n2) Then
            Console.WriteLine("{0} es menor que {1}", n1, n2)
        End If
        Console.Read()
    End Sub
End Module
```

Programa 3.Cálculo de potencia.

```
Module potencia
    Sub Main()
        Dim base, exponente, resultado As Integer
        Console.Write("Ingresa una base entera: ")
        base = Console.ReadLine()
        Console.Write("Ingresa un exponente entero: ")
        exponente = Console.ReadLine()
        resultado = (base ^ exponente)
        Console.WriteLine("{0} elevado al {1} es igual a {2}", _
            base, exponente, resultado)
        Console.Read()
    End Sub
End Module
```

Programa 4. Estructura While. Promedio de 5 números.

```
Module promedio
    Sub Main()
        Dim total As Integer = 0
        Dim contador As Integer = 1
        Dim numero As Integer
        Dim promedio As Double
        While contador <= 5
            Console.Write("Ingresa un número: ")
            numero = Console.ReadLine()
            total += numero
            contador += 1
        End While
        promedio = total / 5
        Console.WriteLine()
        Console.WriteLine("El promedio es {0}", promedio)
        Console.Read()
    End Sub
End Module
```

Programa 5. Estructuras Do While...Loop y Do Until...Loop. Impresión de un cuadrado.

```
Module cuadrado
    Sub Main()
        Dim lado, columna As Integer
        Dim renglon As Integer = 1
        Console.Write("Ingresa el largo (menor o igual a 20): ")
        lado = Console.ReadLine()
        If lado <= 20 Then
            Do While renglon <= lado
                columna = 1
                Do Until columna > lado
                    Console.Write("* ")
                    columna += 1
                Loop
                Console.WriteLine()
                renglon += 1
            Loop
        Else
            Console.WriteLine("Lado demasiado largo")
        End If
        Console.Read()
    End Sub
End Module
```

Programa 6. Estructura de Selección Múltiple

```
Module grado
    Sub Main()
        Dim grado As Integer = 0
        Dim a_cuenta As Integer = 0
        Dim b_cuenta As Integer = 0
        Dim c_cuenta As Integer = 0
        Console.Write("Ingresa un grado, -1 para salir: ")
        grado = Console.ReadLine()
```

```

While grado <> -1
    Select Case grado
        Case 50 To 100
            Console.WriteLine("Letra de grado: A" & vbCrLf)
            a_cuenta += 1
        Case 20, 31 To 49
            Console.WriteLine("Letra de grado: B" & vbCrLf)
            b_cuenta += 1
        Case 0, 10, 15
            Console.WriteLine("Letra de grado: C" & vbCrLf)
            c_cuenta += 1
        Case Else
            Console.WriteLine("Entrada no valida." & _
                "por favor ingresa un grado válido" & vbCrLf)
    End Select
    Console.Write("Ingresa un grado, -1 para quitar: ")
    grado = Console.ReadLine()
End While
Console.WriteLine("Los totales para cada letra de grado son: " & vbCrLf & _
    "A: " & a_cuenta & vbCrLf & "B: " & b_cuenta & _
    & vbCrLf & "C: " & c_cuenta & vbCrLf)
Console.Read()
End Sub
End Module

```

Programa 7. Imprime una lista de precios de artículos a un precio unitario de \$10 cada artículo. Hacerla con 50 artículos en donde se otorgue un descuento del 10% si el cliente compra de 20 a 30 artículos y un 20% de descuento si compra más de 30 artículos. Emplea Do While...Loop. Observa la posible salida:

Artículos	Importe
1	10
2	20
.....	
20	180
21	189
.....	
40	320

Estructura For/Next

La estructura de repetición For/Next ejecuta un conjunto de instrucciones a través de una variable de tipo contador, revisa el siguiente ejemplo:

```

For contador= 1 to 10
    Console.write(contador & " ")
Next

```

La estructura For/Next se utiliza cuando se sabe con certeza el número de veces que las instrucciones se deben de ejecutar.

Estructura Do/Loop While

Esta estructura es muy similar a **While/End While** y **Do While/Loop**, en las cuáles la condición se ejecuta al principio del cuerpo de la estructura, pero en **Do/Loop While** primero se ejecuta el cuerpo de instrucciones y luego se evalúa la condición, lo que permite ejecutar el ciclo por lo menos una vez, observa el siguiente ejemplo:

```
Dim contador As Integer = 1
Do
    Console.Write(contador & " ")
    Contador += 1
Loop While contador <= 5
```

Hay que hacer notar, que la variable *contador* se debe de inicializar con un valor antes de entrar a la estructura, y dentro de esta, incrementar dicho contador para ejecutar el ciclo de instrucciones mientras la condición sea verdadera.

Estructura Do/Loop Until

Es una estructura similar a la estructura **Do Until/Loop** excepto que la condición se evalúa al final y por lo menos las instrucciones se ejecutan una vez. El ciclo se repite siempre y cuando la condición resulte falsa. La sintaxis la observas en el siguiente ejemplo:

```
Dim contador As Integer = 1
Do
    Console.Write(contador & " ")
    Contador += 1
Loop Until contador > 5
```

Este ejemplo hace lo mismo que el anterior de Do/Loop While, nota que aparte de su sintaxis, difiere en el operador de la condición, porque los ciclos de la estructura anterior continúan siempre que la condición sea verdadera y la segunda siempre que la condición sea falsa.

Clase MessageBox

Esta clase requiere de la referencia a librería System.Windows.Forms que la contiene entre otras muchas. Su función es mostrar un cuadro de dialogo dentro del cual se pueden colocar textos, botones e iconos. Su sintaxis es la siguiente:

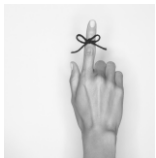
MessageBox.show(variable string, variable string, messageboxbuttons.ok, messageboxicon.information)

Los dos primeros argumentos muestra el contenido de la caja de dialogo y el titulo de la misma, el siguiente indica el tipo de botón que va a incluir y el cuarto argumento indica el icono que aparecerá a la izquierda del mensaje. En las siguientes tablas se indican los tipos de botones e iconos que se pueden utilizar:

Sintaxis	Descripción
MessageBoxIcon.Exclamation	Icono que contiene una exclamación. Indica al usuario que tenga precaución sobre problemas potenciales.
MessageBoxIcon.Information	Icono que contiene una letra "i" dentro de un globo. Indica

	información acerca del estado de la aplicación.
MessageBoxIcon.Question	Icono que contiene un símbolo de interrogación. Usado para preguntar al usuario una cuestión.
MessageBoxIcon.Error	Icono que contiene una X dentro de un círculo rojo. Utilizado para alertar al usuario de situaciones críticas.

Sintaxis	Descripción
MessageBoxButtons.OK	Permite al usuario aceptar un mensaje
MessageBoxButtons.OKCancel	Se continúa o se cancela una operación
MessageBoxButtons.YesNo	Permite al usuario responder a una pregunta
MessageBoxButtons.YesNoCancel	Permite responder una pregunta o cancelar la operación
MessageBoxButtons.RetryCancel	Permite reintentar o cancelar una operación que previamente ha fallado.
MessageBoxButtons.AbortRetryIgnore	Cuando una de una serie de operaciones ha fallado, permiten al usuario abortar toda la secuencia, reintentar la operación fallida o ignorar el fallo y continuar.



Para poder adicionar la librería System.Windows.Forms y poder utilizar la clase MessageBox se debe de proceder como sigue: Menu **Proyecto / Agregar Referencia...** / en la **ficha Net** buscar y dar clic sobre el componente **System.Windows.Forms** y pulsar el botón de **Aceptar**. En el explorador de soluciones activar el botón de *Mostrar todos los archivos* y en la opción de *References* da clic sobre el símbolo de "+" para observar que la librería ya ha sido añadida.

Programa 8. Calcular a interés compuesto el valor futuro de una inversión de \$1,000.00 pesos con una tasa de interés anual del 5% invertido durante 10 años. Capture y ejecute el siguiente programa. Utilizando la estructura For/Next y exhibiendo los resultados en una caja de dialogo en donde se muestre el valor futuro por cada año de la inversión.

```
Imports System.Windows.Forms

Module interes_compuesto
    Sub Main()
        Dim importe, principal As Decimal
        Dim tasa As Double
        Dim periodo As Integer
        Dim salida As String
        principal = 1000.0
        tasa = 0.05
        salida = "Año" & vbTab & "Importe y deposito" & vbCrLf
        For periodo = 1 To 10
            importe = principal * (1 + tasa) ^ periodo
            salida &= periodo & vbTab _
                & String.Format("{0:c}", importe) & vbCrLf
        Next
        messagebox.show(salida, "Interes compuesto", _
            messageboxbuttons.ok, messageboxicon.information)
```

```
End Sub  
End Module
```

Programa 9. Escribe un programa que muestre en una caja de dialogo la suma acumulada de los números de 1 al 50 con la leyenda “La suma es: ” y como titulo de la ventana “Suma de enteros del 1 al 50”. Utiliza la estructura **Do/Loop While**.

Programa 10. Usando la estructura **Do/Loop Until** elabora un programa que muestre en un cuadro de dialogo cuyo titulo de la ventana indique “Series de Números” lo siguiente:

N	N*2	N*4
2	4	8
4	8	16
6	12	24
8	16	32

La estructura deberá de ejecutar cuando menos 20 ciclos.