25 Ejercicios Resueltos Informatica

Informática para a Enxeñaría (Universidade de Vigo)

IPI. Ejercicios resueltos

Tabla de contenidos

1 Ejercicio de bucles (grado de dificultad: 0)

1.1 Enunciado

Calcular el dinero que se acumula tras un número de años, sabiendo que cada año se incrementa dicho capital en un porcentaje (tipo de interés).

1.2 Solución

```
Option Explicit On   ' nos obliga a declarar las variables
Module Module1
    Sub Main()

        Dim inicial As Single, interes As Single
        Dim acumulado As Single, plazo As Integer
        Dim i As Integer, incremento As Single

        ' como los intereses y el dinero pueden tener decimales,
        ' se usa Single (y no Integer o Long)

        ' Preguntamos al usuario los valores necesarios para el cálculo
        inicial = InputBox("¿Capital inicial?")
        interes = InputBox("¿Tipo de interés (%)?")
        plazo = InputBox("¿Período en años?")

        acumulado = inicial
        For i = 1 To plazo  ' para cada año se hace el siguiente cálculo:

            ' cada año se incrementa el dinero:
            incremento = acumulado * interes / 100 ' dividido por 100 pq es %

            ' se acumula a lo ya disponible
            acumulado = acumulado + incremento
        Next i

        MsgBox("El capital acumulado es de " & acumulado)
    End Sub
End Module
```

2 Reloj (grado de dificultad: 0)

2.1 Enunciado

Alcance: constantes y variables, tipos de datos y operaciones aritméticas básicas.

Las agujas que marcan la hora y el minuto en un reloj analógico se superponen a las 0:00:00 h, cual es el siguiente instante en el que se vuelven a superponer (es un poco después de la 1:05:00 h).

El programa deberá:

- Definir constantes y variables.
- Calcular la hora.
- Mostrar el resultado en formato h:m:s.

2.2 Solución

```
Module Module1
    Sub Main()
        'Justificación
        ' angh = hora*gradoshh
        ' angm =(hora-1)*gradoshm
        ' angh=angm
        ' h = gradoshm/(gradoshm-gradoshh)
        Const gradoshh = 360.0 / 12.0
        Const gradoshm = 360.0
        Dim h As Double = gradoshm / (gradoshm - gradoshh)
        Dim hora, minuto, segundo As Integer
        hora = h ' Convierte a entero, despreciando los decimales
        minuto = (h - hora) * 60.0
        segundo = (h - hora) * 3600.0 - minuto * 60.0
        Console.WriteLine("Hora: " & hora & ":" & minuto & ":" & segundo)
        Console.ReadLine()
    End Sub
End Module
```

3 Balística (grado de dificultad: 0)

3.1 Enunciado

Alcance: constantes y variables y operaciones matemáticas básicas.

Desarrollar un programa capaz de calcular el tiempo y distancia que tarda un proyectil en impactar con el suelo, dadas una velocidad y ángulo de disparo.

El programa deberá:

- Definir constantes (gravedad) y variables.
- Solicitar al usuario la velocidad y el ángulo de salida.
- Calcular el tiempo y distancia.
- Mostrar al usuario los datos anteriormente calculados.

3.2 Solución

```
Module Module1
    Sub Main()
        Const g = 9.81
        Dim Vo, ang As Double
        Dim tiempo, distancia As Double

        Console.Write("Velocidad (m/s): ")
        Vo = Console.ReadLine()
        Console.Write("Angulo (grados): ")
        ang = (Math.PI / 180.0) * Console.ReadLine() 'convertir de grados a radianes
        tiempo = Vo * Math.Sin(ang) / (0.5 * g)
        distancia = Vo * Math.Cos(ang) * tiempo

        Console.WriteLine("Tiempo: " & FormatNumber(tiempo, 2) & " s")
        Console.WriteLine("Distancia: " & FormatNumber(distancia, 2) & " m")
        Console.ReadLine()
    End Sub
End Module
```

4 Cálculo salarial (grado de dificultad: 1)

4.1 Enunciado

Alcance: constantes y variables y operaciones aritméticas básicas.

Cuando una empresa paga a sus empleados por su trabajo, el estado le obliga a retener una cantidad en concepto de impuestos que paga el empleado y además abonarle una cantidad en concepto de los impuestos que paga la empresa. Se entiende como "salario bruto" la cantidad que cobra el empleado anualmente antes de que la empresa retenga los impuestos que le corresponden pagar al empleado, "salario neto la cantidad que recibe el empleado" y "coste del empleado" al salario bruto más los impuestos que ha de pagar la empresa (a mayores) por el empleado. Además la empresa ha de tener en cuenta que si en algún momento despide a un empleado tiene que indemnizarle con un determinado número de días de salario por año trabajado, de manera que algunas empresas contemplan una provisión por despido y lo consideran un coste más asociado al empleado. Además, si la empresa tiene beneficios tendrá que pagar un impuesto sobre los mismos. En España, los principales costes son (valores aproximados y simplificado):

Los que la empresa le retiene al trabajador o este paga:

- Cuota de Seguridad Social del Obrero: 6,4% del salario bruto
- IRPF (impuesto de la Renta de las Personas Físicas): 20% del salario bruto
- IVA (Impuesto de Valor Añadido): 21% de los gastos

Los que la empresa paga a mayores:

- Cuota de Seguridad Social de la Empresa: 36,22% del salario bruto
- Provisión de Indemnización: 30 días por año trabajado (365 días)
- Impuesto de Sociedades: 25% del beneficio (ingresos-gastos)

Se supondrá que todos los empleados de la empresa cobran lo mismo, trabajan 1800 horas al año, reciben 12 pagas mensuales meses y que la empresa tiene unos gastos generales (alquileres, luz…) que son el 15% de sus ingresos.

El programa deberá:

- Definir constantes y variables.
- Solicitar al usuario el salario neto, el número de empleados y el precio por hora que le factura a sus clientes por el trabajo de los empleados.
- Calcular ingresos, coste salarial, impuestos directos, beneficios después de impuestos y el poder adquisitivo real de los empleados sobre su coste total (gastan el 60% de su salario neto, sobre lo que pagan IVA).
- Mostrar al usuario los datos anteriormente calculados.

4.2 Solución

```
Module Module1
```

```
    Sub Main()
        ' Constantes
        Const CSSE = 0.3622     'Cuota Seguridad Social de la empresa
        Const CSSO = 0.064      'Cuota Seguridad Social del obrero
        Const IRPF = 0.2        'Impuesto sobre la Renta
        Const PIND = 30.0 / 365 'Provisión Indenmización
        Const ISOC = 0.25       'Impuesto de socidedades
        Const IVA = 0.21        'Impuesto de Valor Añadido
        Const CGG = 0.15        'Coeficiente de Gastos Generales
        Const CGE = 0.6         'Coeficiente de Gasto de los Empleados
        Const meses = 12
        Const horasanho = 1800

        ' Variables
        Dim salarioNeto, empleados, preciohora As Double
        Dim salarioBruto, costeEmpleado, impSociedades As Double
        Dim costeSalarial, costes, ingresos, beneficios As Double
        Dim impDirectos, impIndirectos, impTotales As Double
        Dim salarioDisponible, poderAdquisitivo As Double

        ' Pedir datos al usuario
        Console.Write("¿Salario Neto Mensual?: ")
        salarioNeto = Console.ReadLine()
        Console.Write("¿Número de empleados?: ")
        empleados = Console.ReadLine()
        Console.Write("¿Precio hora?: ")
        preciohora = Console.ReadLine()

        'Hacer cálculos
        salarioBruto = salarioNeto / ((1 - CSSO) * (1 - IRPF))
        costeEmpleado = salarioBruto * (1 + CSSE) * (1 + PIND)
        costeSalarial = costeEmpleado * empleados * meses
        ingresos = empleados * horasanho * preciohora
        costes = costeSalarial + CGG * ingresos
        beneficios = (ingresos - costes) 'antes de impuestos
        impSociedades = beneficios * (1 - ISOC) 'Suponemos beneficios >0
        beneficios -= impSociedades 'después de impuestos
        impDirectos = (costeEmpleado - salarioNeto) * empleados * meses +
impSociedades
        impIndirectos = salarioNeto * empleados * meses * CGE * IVA
        impTotales = impDirectos + impIndirectos
        salarioDisponible = salarioNeto * empleados * meses - impIndirectos
        poderAdquisitivo = salarioDisponible / costeSalarial

        'Mostrar resultados
        Console.WriteLine("Ingresos: " & FormatNumber(ingresos, 2))
        Console.WriteLine("Coste Salarial: " & FormatNumber(costeSalarial, 2))
        Console.WriteLine("Impuestos Directos: " & FormatNumber(impDirectos, 2))
        Console.WriteLine("Beneficios: " & FormatNumber(beneficios, 2))
        Console.WriteLine("Poder Adquistivo: " & FormatNumber(poderAdquisitivo, 2))
        Console.ReadLine() 'Esperar a que el usuario pulse una tecla
    End Sub
End Module
```

5 Ejercicio de vectores (grado de dificultad: 1)

5.1 Enunciado

Dada un vector que representa las temperaturas medidas durante un período de tiempo, localizar la variación máxima de temperatura (diferencia entre los valores máximo y mínimo)

5.2 Solución

```
Option Explicit On
Module Module1
    Sub Main()
        ' Dada un vector que representa las temperaturas medidas
        ' durante un período de tiempo, localizar la variación máxima
        ' de temperatura(diferencia entre los valores máximo y mínimo)
        Dim numpos As Integer, i As Integer
        ' ojo: las temperaturas pueden tener decimales, por lo para ellas se
        ' debe usar Single o Double:
        Dim v() As Single ' paréntesis vacíos pq aún no sabemo el nº de posiciones
        Dim max As Single, min As Single

        ' CARGA DE DATOS EN EL VECTOR
        ' preguntamos al usuario el número de posiciones del vector
        numpos = InputBox("¿Cuántos valores se guardarán?")
        ' creamos el vector
        ReDim v(numpos - 1) ' entre paréntesis: nº de pos (siempre empiezan en cero)
        ' pedimos los datos y los guardamos en el vector
        For i = 0 To numpos - 1 ' recorremos todas las posiciones del vector (empieza en 0)
            v(i) = InputBox("¿Valor de la posición " & i & " ?")
        Next i
        ' BÚSQUEDA DE LOS VALORES MÁXIMO Y MÍNIMO
        For i = 0 To numpos - 1 ' recorremos valor por valor
            ' si es el primer valor analizado (i=0) o es menor q
            ' el que hasta ese momento era el máximo (max):
            If (i = 0) Or v(i) > max Then
                max = v(i)
            End If
            ' de modo similar para el mínimo
            If (i = 0) Or v(i) < min Then
                min = v(i)
            End If
        Next
        ' MOSTRAMOS EL RESULTADO:
        MsgBox("La máxima variación de temperatura es de " & max - min & " grados")
    End Sub
End Module
```

6 Cálculo del mínimo de una matriz 2D (grado de dificultad: 1)

6.1 Enunciado

Cálculo del mínimo de una matriz 2D.

6.2 Solución

```
Option Explicit On
Module EjemploMinimo
  Const NF = 3   ' constante: número de filas
  Const NC = 2   ' constante: número de columnas
  Sub main()
    Dim A(NF - 1, NC - 1) As Integer  ' crea la matriz de enteros (los índices
empiezan en cero)
    Dim i As Integer, j As Integer, menor As Integer
    For i = 0 To NF - 1   ' pide los datos de la matriz
      For j = 0 To NC - 1
        A(i, j) = InputBox("A(" & i & "," & j & ")")
      Next j
    Next i
    menor = A(0, 0)    ' inicializa: de momento, el menor es el primero
    For i = 0 To NF - 1
      For j = 0 To NC - 1
        If (menor > A(i, j)) Then menor = A(i, j) ' si hay alguno menor,
actualiza "menor"
      Next j
    Next i
  End Sub
End Module
' NOTA: si no se inicializa la varible "menor", en caso de
' que los valores de la matriz sean negativos, no funcionaría
' el programa
```

7 Definición y uso de una función que indique si un valor es par o impar (grado de dificultad: 1)

7.1 Enunciado

Definición y uso de una función que indique si un valor es par o impar.

7.2 Solución

```
Option Explicit On
Module EjemploMinimo
  Sub main()
    Dim i As Integer, esPar As Boolean
    i = InputBox("Introduce un número")
    esPar = par(i)        ' llamamos a la función pasándole "i" y recogiendo el
resultado
    If (esPar = True) Then      ' en función del resultado mostramos el mensaje
correspondiente
      MsgBox("EL NÚMERO " & i & " ES PAR")
    Else
      MsgBox("EL NÚMERO " & i & " ES IMPAR")
    End If
  End Sub

  ' definición de la función
  Function par(ByVal n As Integer) As Boolean  ' función que recibe un entero y
devuelve un boolean
    Dim resultado As Boolean
    If (n Mod 2 = 0) Then  ' si el resto de dividir por dos es cero, el número
es par
      resultado = True
    Else
      resultado = False
    End If
    Return resultado     ' devuelve el resultado
  End Function
End Module
```

8 Transformar valor entero positivo en binario (grado de dificultad: 1)

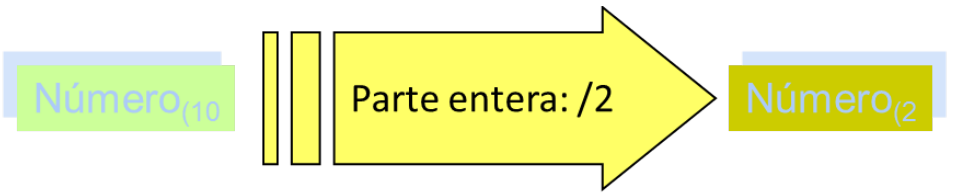
8.1 Enunciado

Realizar un programa en Visual Basic que solicite un valor entero positivo al usuario y muestre:

- Su valor correspondiente en binario con el siguiente mensaje: "El valor en binario del número XXX es: YYYYYYYYYY"
- El número de bits que tiene dicho valor binario con el siguiente mensaje: "El número de bits que ocupa son: ZZZZZZ"

Para pasar un valor entero a binario se realizarán sucesivas divisiones por 2, empezando por el número y continuando por el cociente entero hasta que el cociente entero sea 0. El valor binario estaría formado por los restos de cada una de las divisiones sucesivas que se van haciendo empezando por el último resto calculado.

- Resto de una división utilizar la expresión VALOR MOD 2; esta operación devuelve el resto de dicha división
- Cociente entero de una división utilizar la expresión int(VALOR / 2). Esta función Int() devuelve el valor entero del cociente.



Parte entera
11 / 2 = 5 Resto **1**
5 / 2 = 2 Resto **1**
2 / 2 = 1 Resto **0**
1 / 2 = 0 Resto **1**
(finaliza cuando el cociente es 0)

8.2 Solución

```
Module Module1

  Sub main()
    'Definimos la variables a utilizar
    Const base As Byte = 2 'Vble que almacena el cambio de base a binario
    Dim numero, contar_bits, cociente As Integer
    Dim Resto As Byte  'Vble que guarda el resto de la operación MOD
    Dim respuesta As Char 'Vble para guardar la respuesta de repetir el programa
    Dim valor_binario As String  'Vble para ir guardando el numero binario

    Console.WriteLine("Calcular numero binario")
    Do
      contar_bits = 0   'Inicializamos el contador de bits a cero
```

```
      valor_binario = ""  'Inicializamos la cadena a "" (Vacio)
      Do  'repito petición de número hasta que sea >0
        numero = InputBox("Dame un número?")
        If numero < 0 Then
          MsgBox("ERRROR: debe ser un valor positivo")
        End If
      Loop Until numero > 0
      'igualo el cociente al número para empezar a dividir por el mismo
      'y luego ir cambiando el valor a medida que se divide, asi mantenemos
      'el valor del número introducido sin modificarlo
      cociente = numero
      Do
        Resto = cociente Mod base
        'almaceno el resto en la cadena valor_binario
        'recordar que empieza de izquierda a derecha la secuencia del binario
        'de ahí el orden en la concatenación con el operador &
        valor_binario = Resto & " " & valor_binario
        cociente = Int(cociente / base)
        contar_bits = contar_bits + 1  'contabilizamos las veces que se hace la
operación
      Loop While cociente > 0
      'visualizamos las soluciones
      MsgBox("El valor en base (" & base & ") del número " & numero & " es: " &
valor_binario)
      MsgBox("el número de bits creados son: " & contar_bits)
      'Preguntamos si desea volver a ejecutar el programa
      Do
        respuesta = UCase(InputBox("Desea ejecutar de nuevo el programa? (s/n)
"))
        If respuesta <> "S" Then 'se podría empezar por comparar el valor "N"
          If respuesta <> "N" Then
            MsgBox("ERROR: respuesta no válida. Debe pulsar 'S' o 'N'")
          End If
        End If
      Loop Until respuesta = "S" Or respuesta = "N"
    Loop While respuesta <> "N"
  End Sub

End Module
```

9 Ejercicio de funciones (grado de dificultad: 2)

9.1 Enunciado

Codificar una función que devuelva la suma de una lista de valores reales recibida como un argumento de tipo vector. Probar la función en un programa (Sub Main).

9.2 Solución

```
Option Explicit On
Module Module1
    Sub Main()
        ' crear un vector dándole los valores (no es necesario
        ' indicar el número de elementos en la declaración)
        Dim valores() As Double = {1.5, 4.5, 2, 2}
        Dim suma As Double

        ' invocamos la función y recogemos el resultado
        ' (nótese que los nombres del argumento no tiene que
        ' coincidir con el vector que se pasa en la llamada)
        suma = sumatorio(valores)
        MsgBox("Suma: " & suma)
    End Sub

    ' la función se declara dentro del módulo, pero
    ' fuera del Sub Main() y de cualquier otra
    ' subrutina del módulo
    Function sumatorio(v() As Double) As Double
        Dim i As Integer, limiteSup As Integer
        Dim resultado As Double = 0

        ' calculamos el número de posiciones del vector
        limiteSup = UBound(v, 1)

        For i = 0 To limiteSup ' recorremos el vector posición por posición
            resultado += v(i)  ' acumulamos el valor de cada posición
        Next i
        Return resultado  ' devolvemos el resultado
    End Function
End Module
```

10 Dividir un número (grado de dificultad: 2)

10.1Enunciado

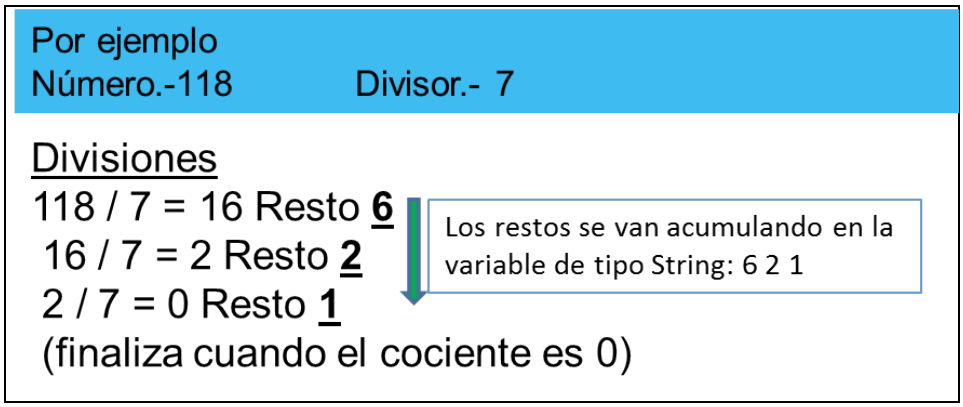
Realizar un programa en Visual Basic que solicite un valor entero positivo (numero) al usuario y un valor entero positivo (divisor) por el cual dividirlo.

Teniendo en cuanta que si el valor por el cual se divide es mayor al número, deberá volver a pedir dicho valor, el programa deberá realizar sucesivas divisiones del valor entero entre su divisor, empezando por el número y continuando por el cociente entero hasta que el cociente entero sea 0. También debe ir guardando en una variable tipo String los sucesivos restos de las divisiones que se vayan realizando separados por un espacio. A continuación debe mostrar:

- El valor de la variable String que contiene los restos con el siguiente mensaje: "Los restos de las divisiones del número XXX son: YYYYYYYYYY"
- El número de divisiones que se realizaron con el siguiente mensaje: "El número de divisiones realizadas son: ZZZZZZ"

NOTA.-

- Resto de una división utilizar la expresión NUMERO MOD DIVISOR; esta operación devuelve el resto de dicha división
- Cociente entero de una división utilizar la expresión int(NUMERO / DIVISOR). La función INT(expresión) devuelve el valor entero del resultado de la expresión o el valor sin decimales).



10.2Solución

```
Module Module1
  Sub main()
    'Definimos la variables a utilizar
    Dim numero, divisor, contar_divisiones, cociente As Integer
    Dim Resto As Byte  'Vble que guarda el resto de la operación MOD
    Dim respuesta As Char 'Vble para guardar la respuesta de repetir el programa
    Dim valor_restos As String  'Vble para ir guardando los restos de las
divisiones

    Console.WriteLine("Divisiones sucesivas")
    Do
      contar_divisiones = 0 'Inicializamos el contador de divisiones a cero
      valor_restos = "" 'Inicializamos la cadena de restos a "" (vacio)

      Do 'Pedimos el numero hasta que sea positivo
        numero = InputBox("Dame un número?")
        If numero < 0 Then
```

```
          MsgBox("ERRROR: debe ser un valor positivo")
        End If
      Loop Until numero > 0
      Do 'Pedimos el divisor hasta que sea positivo y mayor que el numero
        divisor = InputBox("dame su divisor?")
        If (divisor > numero) Or (divisor < 0) Then
          MsgBox("ERROR: el divisor tiene que ser menor que " & numero & " y
mayor que cero.")
        End If
      Loop Until divisor < numero And divisor > 0
      'igualo el cociente al número para empezar a dividir por el mismo
      'y luego ir cambiando el valor a medida que se divide, asi mantenemos
      'el valor del número introducido sin modificarlo
      cociente = numero
      Do
        Resto = cociente Mod divisor
        'almaceno el resto en la cadena valor_resto
        valor_restos = valor_restos & " " & Resto
        cociente = Int(cociente / divisor)
        contar_divisiones = contar_divisiones + 1 'contabilizamos las veces que
se hace la operación
      Loop While cociente > 0
      'Visualizamos las soluciones
      MsgBox("Los restos de las divisiones sucesivas del número " & numero & "
son: " & valor_restos)
      MsgBox("El número de divisiones realizadas son: " & contar_divisiones)
      'Preguntamos si desea volver a ejecutar el programa
      Do
        respuesta = UCase(InputBox("Desea ejecutar de nuevo el programa? (s/n)
"))
        If respuesta <> "S" Then 'se podría empezar por comparar el valor "N"
          If respuesta <> "N" Then
            MsgBox("ERROR: respuesta no válida. Debe pulsar 'S' o 'N'")
          End If
        End If
      Loop Until respuesta = "S" Or respuesta = "N"
    Loop While respuesta <> "N"
  End Sub

End Module
```

11 Transformar un valor binario en un número entero positivo (grado de dificultad: 2)

11.1Enunciado

Realizar un programa en VB que le pida un número en binario al usuario (máximo 7 dígitos), cuyos valores de forma individual serán almacenados en un vector de tipo Byte y que sólo pueden ser "0" (CERO) o "1" (UNO). Se le solicitará al usuario cuantos dígitos en binario va a introducir (recordando que como máximo sólo puede meter 7). Entre las variables utilizadas debe existir una variable constante llamada "BASE" con el valor=2. A continuación el programa debe mostrar:

- El valor correspondiente en formato decimal, mediante el siguiente mensaje. "El valor en decimal es: XXXXXXX"

Para ello hay que definir en el programa una función que se le pase el vector con los datos binarios y devuelva el valor entero del número decimal obtenido.

Ejemplo del cálculo a realizar:

$$\text{Ej.- } 110_{(2} = \underline{1}*2^2 + \underline{1}*2^1 + \underline{0}*2^0 = 4 + 2 + 0 = 6_{(10}$$

Nota.- La solución del cálculo es multiplicar cada dígito por la base (en este caso 2) elevada a la posición que ocupa el dígito. Cada dígito binario ocupa una posición, que siempre empieza de derecha a izquierda desde la posición cero hasta el número de dígitos.
IMPORTANTE.- Al introducir los dígitos, se empieza siempre por la posición mayor.

11.2Solución

```
Module Module1
  Sub main()
    Dim binario() As Byte 'vble para almacenar los dígitos binarios
    ' vble i para recorrer el indice el vector
    ' vble j para mostrar al usuario el orden al reves de como los introduce
    '(se podría no declarar, es una cuestión de visualizar al usuario que valor
    'está introduciendo y en que orden)
    'vble ctos_digitos para almacenar el total de digítos que tiene el valor
binario
    Dim i, j, ctos_digitos As Integer

    Console.WriteLine("Transformar un número binario a decimal")
    'pedimos cuantos digitos binarios va a introducir
    Do
      Console.WriteLine("Cuantos dígitos binarios va a introducir
(0>cantidad<=7?")
      ctos_digitos = Console.ReadLine()
      'comprobamos que sea mayor a cero y menor/igual a 7
      If (ctos_digitos > 7 Or ctos_digitos <= 0) Then
```

```
        Console.WriteLine("ERROR.- el valor debe ser mayor que cero o
igual/menor que 7")
      End If
    Loop Until (ctos_digitos <= 7 Or ctos_digitos > 0)
    'Creamos el vector con el número de digitos que ha
    'indicado el usuario menos uno
    ReDim binario(ctos_digitos - 1)
    'inicializamos j a uno para que le indique al usuario que
    'valor está introduciendo, aunque se almacene al revés
    j = 1
    'solicitamos cada uno de los dígitos al usuario
    'almacenandolos desde la posición mayor hasta la posición cero
    For i = UBound(binario) To 0 Step -1
      Do
        Console.Write("Introduzca el valor binario (" & j & ")= ")
        binario(i) = Console.ReadLine()
        If (binario(i) < 0) Or (binario(i) > 1) Then
          Console.WriteLine("ERROR: el valor sólo puede ser ""0"" (cero) o ""1""
(uno)")
        End If
      Loop Until (binario(i) = 0 Or binario(i) = 1)
      'vamos incrementando j para que le aparezca al usuario el valor a
introducir correcto
      j = j + 1
    Next
    'llamamos a la función binario_decimal que devuelve el valor en binario
    Console.WriteLine("el valor en decimal es= " & binario_decimal(binario))
    Console.ReadLine()
  End Sub
  'Función que se le pasa el vector en binario y devuelve el valor en decimal
  Function binario_decimal(ByVal vector_binario() As Byte) As Integer
    Dim i As Integer
    Dim valor_decimal As Integer 'para almacenar el calculo en binario
    Const base As Byte = 2
    'realizamos el calculo
    'se podría empezar desde cero hasta el ultimo y daría igual
    For i = UBound(vector_binario) To 0 Step -1
      valor_decimal = valor_decimal + (vector_binario(i) * base ^ i)
    Next
    'devolvemos el valor decimal calculado
    Return valor_decimal
  End Function

End Module
```

12 Control de asistencias (grado de dificultad: 2)

12.1Enunciado

Diseñar un programa en VB en modo console que permita gestionar las asistencias a clases de teoría y prácticas de los 5 grupos (los grupos son A-B-C-D-E) de la asignatura Informática para ingeniería en una determinada semana. Los datos se almacenarán en una matriz que guarde en la primera fila las asistencias a teoría (el máximo número de asistentes será de 100 alumnos) y en la segunda fila las asistencias a prácticas (el máximo número de asistentes será de 25 alumnos). Una vez introducidos los datos se mostrará el siguiente menú:

1. Indicar el promedio de asistencias a teoría
2. Indicar que grupos (A-B-C-D-E) tienen una asistencia a prácticas inferior a 10 alumnos (utilizar un string para visualizarlo). NOTA.- Si no existe ninguno, indicarlo con el mensaje "No existe ningún grupo con asistencia menor a 10 alumnos"
3. Visualizar todos los datos de asistencias de todos los grupos tanto de teoría como prácticas (en un string).
4. Salir.

12.2Solución

```
Module Module1
    Sub Main()
        'Declaración de variables
        Dim matriz(1, 4) As Byte
        Dim grupo() As String = {"A", "B", "C", "D", "E"}
        Dim horarios() As String = {"Teoría", "Práctica"}
        Dim Horario_mayor, cantidad As Integer
        Dim matriz_resultado As String = ""
        Dim repetidos As String = ""
        Dim matriz_promedio As String = ""
        Dim promedio As Double
        Dim existe As Integer
        Dim i, j, opcion As Byte
        'Pide asistencias
        For i = 0 To 1
            For j = 0 To 4
                Do

                    matriz(i, j) = InputBox("dame las asistencias: " & grupo(j) & "  " &
horarios(i))

                    If i = 0 Then
                        cantidad = 125
                    Else
                        cantidad = 25
                    End If
                    'Verificar el número de alumnos que los que se nos permiten
                    If matriz(i, j) > cantidad Then
                        MsgBox("no puede haber en el horario: " & horarios(i) & " más que "
& cantidad)
                    End If
                Loop Until matriz(i, j) < cantidad
            Next
        Next

        'Crear el menú
        Do
```

```
            opcion = InputBox("Elige la opción:" & vbLf & "1. Indicar el promedio de
asistencias a teoría" & vbLf & "2. Visualizar si hay grupos de practica con asistencia menor
que 10" & vbLf & "3.Visualizar las asistencias" & vbLf & "4.Salir.")
            Select Case opcion
                Case 1
                    'Opción 1. Promedio de asistencia a Teoría
                    promedio = 0
                    For i = 0 To 4

                        promedio = promedio + matriz(0, i)
                    Next

                    promedio = Math.Truncate(promedio / 5)

                    MsgBox("Promedio de asistencias Teoría: " & promedio & " alumnos")


                Case 2
                    'Opción 2. Que grupos tienen asistencia a prácticas inferior que a 10.
                    existe = 0
                    Horario_mayor = 0
                    repetidos = ""
                    For i = 0 To 4
                        If matriz(1, i) < 10 Then
                            repetidos = repetidos & vbTab & grupo(i)
                        End If
                    Next
                    If repetidos = "" Then
                        MsgBox("No existe ningun grupo con menos de 10 asistencias")
                    Else
                        MsgBox("los grupos con más menos de 10 asistencias: " & vbLf &
repetidos)
                    End If
                Case 3
                    'opcion 3. Visualizar los datos de las asistencias.
                    matriz_resultado = "A                B                C
D                E" & vbLf
                    For i = 0 To 1
                        For j = 0 To 4

                            matriz_resultado = matriz_resultado & matriz(i, j) & vbTab
                        Next
                        matriz_resultado = matriz_resultado & vbLf
                    Next
                    MsgBox(matriz_resultado)
                Case 4
                    ' opcion 4. Salir
                    MsgBox("Salir.")
                Case Else
                    'Tecleo un número correcto.
                    MsgBox("no pusiste algo correcto")
            End Select
        Loop Until opcion = 4
    End Sub
End Module
```

13 Distancia y Punto medio de dos puntos (grado de dificultad: 2)

13.1Enunciado

Crear una rutina/función que calcule la distancia entre dos puntos cualesquiera y la muestre al usuario. Imprimir en el main La distancia. $\sqrt{(x-x1)^2+(y-y1)^2}$. El punto medio se calcula como: $pto\ 3 = \frac{(x-x1)}{2}, \frac{(y-y1)}{2}$. Mostrar el punto medio en pantalla. El código final debe incluir:

```
Module Module1
    Structure Punto
        Dim x As Integer
        Dim y As Integer
    End Structure
    Sub Main()
        Dim pto1, pto2 As Punto
```

1. El programa solicitará dos puntos cualesquiera al usuario y almacenará cada uno de ellos en una variable del tipo punto.
2. A continuación se mostrará los puntos introducidos por el usuario.
3. Se crea una función que calcule la distancia media entre esos dos puntos, y se muestra el resultado en el main.
4. Se crea una función que calcule el punto medio y retorne un tercer punto que representa al punto medio. Se muestra en pantalla.

13.2Solución

```
Module Module4
    Structure Punto
        Dim x As Double
        Dim y As Double
    End Structure
    Sub Main()
        'Declaración de las variables que se utilizarán en el main
        Dim punto_1, punto_2, punto_3 As Punto 'Se crean tres variables del tipo punto
        'Se le piden al usuario los dos valores(x,y)que conforman el punto 1
        Console.WriteLine("Dame punto 1 ")
        With punto_1
            .x = Console.ReadLine
            .y = Console.ReadLine
        End With
        'Se le piden al usuario los dos valores(x,y)que conforman el punto 2
        Console.WriteLine("Dame punto 2 ")
        With punto_2
            .x = Console.ReadLine
            .y = Console.ReadLine
        End With

        'Se muestran los punto introducidos por el usuario
        With punto_1
            Console.WriteLine("X = " & .x & " Y = " & .y)
        End With
        With punto_2
            Console.WriteLine("X = " & .x & " Y = " & .y)
```

```
        End With
        'Se llama a la función que calcula la distancia entre dos puntos, y el resultado se
imprime directamente en pantalla.
        Console.WriteLine(DistanciaPuntos(punto_1, punto_2))
        'Se llama a la función que realiza el cáclulo del punto medio, y su valor se igual a
la variable pto3
        punto_3 = PuntoMedio(punto_1, punto_2)
        'Se imprime el valor del punto medio calculado en la función.
        With punto_3
            Console.WriteLine("X = " & .x & " Y = " & .y)
        End With
        Console.WriteLine("Salir")

    End Sub
    'Función que recibe como parámetros de entrada:
    'pto1,pto: puntos cualquiera
    'regresa como valor el cálculo de la distancia con la funciones Math.pow

    Function DistanciaPuntos(ByVal p1 As Punto, ByVal p2 As Punto) As Double
        Dim distancia As Double = 0
        distancia = Math.Sqrt(Math.Pow(p2.x - p1.x, 2) + Math.Pow(p2.y - p1.y, 2))
        DistanciaPuntos = distancia
    End Function
    'Función que recibe como parámetros de entrada:
    'pto1,pto: puntos cualquiera
    'regresa como valor una variable del tipo punto, conteniendo el punto medio.
    Function PuntoMedio(ByVal p1 As Punto, ByVal p2 As Punto) As Punto
        Dim puntoMed As Punto ' Se crea un nuevo punto
        If puntoMed.x = (p2.x + p1.x) <> 0 Then ' Se comprueba que el número sea mayor a
cero para evitar la división entre 0
            puntoMed.x = (p2.x + p1.x) / 2
        Else
            puntoMed.x = 0
        End If
        If puntoMed.y = (p2.y + p1.y) <> 0 Then
            puntoMed.y = (p2.y + p1.y) / 2
        Else
            puntoMed.y = 0
        End If

        PuntoMedio = puntoMed ' se retorna el punto calculado
    End Function
End Module
```

14 Calculo de la Resistencia total de un circuito, su corriente y voltaje (grado de dificultad: 2)

14.1 Enunciado

Se calcula un circuito en Serie y en Paralelo

En el ejercicio se aplican los conocimientos adquiridos hasta la clase 4

Es decir: Definición de variables, Captura de valores, Visualización de valores

Toma de decisiones, Ciclos, conmutadores, vectores, modo consola

14.2 Solución

```
Module Module1

    Sub Main()

        '
        ' Definición de variables
        '
        Dim voltaje, corriente As Double
        Dim i, numresis As Integer
        Dim restot As Double
        Dim resser() As Double
        Dim respar() As Double
        Dim opcion As Byte
        '
        'Ciclo de control
        '
        Do
            '
            'Visualización del menú y captura de la variable de control
            '
            Console.Clear()
            Console.WriteLine("Menú xeral " & vbCrLf & "1)Resistencias en Serie" & vbCrLf &
"2)Resistencias en Paralelo" & vbCrLf & "3)Fin")
            opcion = Console.ReadLine()
            '
            'Resolución de las distintas opciones
            '
            Select Case opcion
                Case 1 'Caso de las resistencias en Serie
                    Console.Clear()
                    Console.Write("Dime el número de resistencias en serie ")
                    numresis = Console.ReadLine()
                    If numresis < 1 Then
                        Console.WriteLine("Error, el número no puede ser menor que 1 ")
                    Else
                        ReDim resser(numresis)
                        For i = 0 To numresis - 1
                            Console.Write("Resistencia " & i + 1 & " Vale = ")
                            resser(i) = Console.ReadLine()
                        Next
                        restot = 0.0
                        For i = 0 To numresis - 1
                            restot = restot + resser(i)
                        Next
```

```
                        Console.WriteLine("Resistencia total do circuito serie = " & restot)
                        Console.ReadLine()
                    End If
                Case 2 'Caso de las resistencias en paralelo
                    Console.Clear()
                    Console.Write("Dime el número de resistencias en paralelo ")
                    numresis = Console.ReadLine()
                    If numresis < 1 Then
                        Console.WriteLine("Error, el número no puede ser menor que 1 ")
                    Else
                        ReDim respar(numresis)
                        For i = 0 To numresis - 1
                            Console.Write("Resistencia " & i + 1 & " Vale = ")
                            respar(i) = Console.ReadLine()
                        Next
                        restot = 0.0
                        For i = 0 To numresis - 1
                            restot = restot + 1 / respar(i)
                        Next
                        restot = 1 / restot
                        Console.WriteLine("Resistencia total do circuito paralelo = " &
restot)
                        Console.ReadLine()
                    End If
                Case 3 'Mensaje de finalización
                    Console.WriteLine("Fin")
                Case Else
                    Console.WriteLine("Opción erronea")
                    Console.ReadLine()
            End Select
        Loop Until opcion = 3

    End Sub

End Module
```

15 Cálculo de un coeficiente binomial, número combinatorio o combinación (grado de dificultad: 2)

15.1Enunciado

El número de formas de escoger *k* elementos a partir de un conjunto de *n*, puede denotarse de varias formas:https://es.wikipedia.org/wiki/Coeficiente_binomial_-_cite_note-2 $C(n,k), n\,C\,k$, C_k^n *o* $\binom{n}{k}$. Así, por ejemplo, *C*(4,2)=6, puesto que hay 6 formas de escoger 2 elementos a partir de un conjunto con 6 elementos. O dicho de otra forma, con un conjunto de 4 elementos, pueden formarse 6 subconjuntos distintos de 2 elementos.

Los números C(*n*,*k*) se conocen como «**coeficientes binomiales**», pero es frecuente referirse a ellos como «**combinaciones de *n* en *k***», o simplemente «***n* en *k***» o «***n* sobre *k***».

Es importante notar que la definición asume implícitamente que *n* y *k* son naturales, y que además *k* no excede a *n*. Podemos definir C(*n*,*k*)=0 si *k*>*n*, puesto que no es posible escoger más elementos que los que tiene el conjunto dado (por tanto hay *cero* formas de hacer la elección). Estas precisiones cobrarán relevancia más adelante cuando se discutan generalizaciones del concepto (por ejemplo, cuando *n* o *k* sean negativos o cuando no sean números enteros).

15.2Solución sin modularidad

```
Module Module1

    Sub main()

        'declaración de variables

        Dim i As Byte
        Dim n, k As Integer
        Dim factorialn As Double = 1 'es necesario inicializar con el valor 1 (elemento
neutro del producto)
        Dim factorialk As Double = 1
        Dim factorialdif As Double = 1

        'resolución del problema

        Do 'comprobación de validez en la solicitud de parámetros para el número
combinatorio
            n = InputBox("Número de elementos del conjunto (ha de ser un número natural): ")
            k = InputBox("Número de elementos para formar subconjuntos (ha de ser un número
natural): ")
        Loop Until (n > 0 And k > 0)
        If n > k Then 'es posible el cálculo del número combinatorio
            For i = 1 To n 'ciclo para el cálculo de n-factorial
                factorialn = factorialn * i
            Next i
            For i = 1 To k 'ciclo para el cálculo de k-factorial
                factorialk = factorialk * i
            Next i
            For i = 1 To n - k 'ciclo para el cálculo del factorial de n-k
                factorialdif = factorialdif * i
            Next i
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es: " &
factorialn / (factorialk * factorialdif))
        Else ' el resultado del número combinatorio es 0 (no es posible formar subconjuntos)
```

```
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es 0")
        End If

    End Sub

End Module
```

15.3Solución mediante el uso de una función

```
Module Module2

    Function factorial(ByVal num As Integer) As Double 'declaración de la función para el
cálculo de un factorial
        Dim i As Byte
        Dim producto As Double = 1

        For i = 1 To num
            producto = producto * i
        Next
        factorial = producto
    End Function

    Sub main()

        'declaración de variables

        Dim n, k As Integer

        'resolución del problema

        Do
            n = InputBox("Número de elementos del conjunto (ha de ser un número natural): ")
            k = InputBox("Número de elementos para formar subconjuntos (ha de ser un número
natural): ")
        Loop Until (n > 0 And k > 0)
        If n > k Then
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es: " &
factorial(n) / (factorial(k) * factorial(n - k))) 'llamadas a la función
        Else
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es 0")
        End If

    End Sub

End Module
```

15.4Solución mediante el uso de una función recursiva

Una función recursiva es aquella que se llama a sí misma para resolverse. Dicho de otra manera, una función recursiva se resuelve con una llamada a sí misma, cambiando el valor de un parámetro en la llamada a la función.

El proceso de llamadas recursivas siempre tiene que acabar en una llamada a la función que se resuelve de manera directa, sin necesidad de invocar de nuevo la función. Esto será siempre necesario, para que llegue un momento que se corten las llamadas reiterativas a la función y no se entre en un bucle infinito de invocaciones.

Veamos su uso en el ejemplo:

```
Module Module3

    Function factorial(ByVal num As Integer) As Double 'declaración de la función para el
cálculo de un factorial

        If num = 1 Then
            factorial = 1
        Else
            factorial = num * factorial(num - 1)
        End If

    End Function

    Sub main()

        'declaración de variables

        Dim n, k As Integer

        'resolución del problema

        Do
            n = InputBox("Número de elementos del conjunto (ha de ser un número natural): ")
            k = InputBox("Número de elementos para formar subconjuntos (ha de ser un número
natural): ")
        Loop Until (n > 0 And k > 0)
        If n > k Then
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es: " &
factorial(n) / (factorial(k) * factorial(n - k)))
        Else
            MsgBox("El resultado del número combinatorio C(" & n & "," & k & ") es 0")
        End If

    End Sub


End Module
```

16 Integrales (grado de dificultad: 2)

16.1Enunciado

Alcance: constantes y variables, operaciones matemáticas básicas y bucles.

Desarrollar un programa que calcule áreas y volúmenes aplicando integrales, tanto en su forma analítica (aplicando fórmulas) como numérica (rebanadas), de las siguientes funciones u objetos:

- Área debajo de la función f(x) = $2x^2+5x+7$, entre dos valores de x (xmin y xmax)
- Área de una circunferencia dado el radio.
- Volumen de un cono dado su longitud y ángulo de apertura en grados.

El programa deberá:

- Definir constantes (gravedad) y variables.
- Solicitar al usuario los parámetros y el número de pasos para los cálculos.
- Hacer los cálculos.
- Mostrar al usuario los resultados.

16.2Solución

```
Module Module1

    Sub Main()
        Dim xmin, xmax, rcirc, loncono, gcono, pasos As Single
        Dim afunc, acirc, acono As Single ' resultados analíticos
        Dim nfunc, ncirc, ncono As Single ' resultados numéricos
        Dim rcono, rdisco, adisco, res, resa As Single ' variables auxiliares
        ' Leer parámetros
        Console.Write("x mínimo: ")
        xmin = Console.ReadLine()
        Console.Write("x máximo: ")
        xmax = Console.ReadLine()
        Console.Write("radio circumferencia: ")
        rcirc = Console.ReadLine()
        Console.Write("longitud cono: ")
        loncono = Console.ReadLine()
        Console.Write("apertura cono (grados): ")
        gcono = Console.ReadLine()
        Console.Write("número de pasos: ")
        pasos = Console.ReadLine()
        ' Hacer cálculos
        ' función
        afunc = (2 / 3) * (xmax ^ 3 - xmin ^ 3) + (5 / 2) * (xmax ^ 2 - xmin ^ 2) +
7 * (xmax - xmin)
        ' solución numérica, aproximación por rectángulos de la altura f(x) y base
res
        res = (xmax - xmin) / pasos
        For x As Double = xmin To xmax Step res
            nfunc += res * (2 * x ^ 2 + 5 * x + 7)
        Next
        ' círcunferencia
        acirc = Math.PI * rcirc ^ 2
        ' solución numérica, aproximación por trángulos (trozos de tarta)
        res = 2 * Math.PI / pasos
```

```
        For a As Double = 0 To 2 * Math.PI Step res
            ncirc += 0.5 * rcirc ^ 2 * Math.Sin(res)
        Next
        ' cono
        rcono = loncono * Math.Sin(gcono * Math.PI / 180.0) 'se convierte el álgulo
a radianes
        acono = (1 / 3.0) * loncono * Math.PI * rcono ^ 2
        ' solución numérica, aproximación por discos a lo largo del cono
        res = loncono / pasos
        For l As Double = 0 To loncono Step res
            rdisco = l * Math.Sin(gcono * Math.PI / 180.0)
            adisco = 0 ' hay que ponerla a cero porque esta variable se va a usar
varias veces
            resa = 2 * Math.PI / 1000 'lo limitamos porque sino hariamos cálculos de
pasos^2
            For a As Double = 0 To 2.0 * Math.PI Step resa
                adisco += 0.5 * rdisco * rdisco * Math.Sin(resa)
            Next
            ncono += res * adisco
        Next
        ' Mostrar resultados
        Console.WriteLine("Area de la función: " & FormatNumber(afunc, 2) & " | " &
FormatNumber(nfunc, 2))
        Console.WriteLine("Area de la circunferencia: " & FormatNumber(acirc, 2) & "
| " & FormatNumber(ncirc, 2))
        Console.WriteLine("Volumen del cono: " & FormatNumber(acono, 2) & " | " &
FormatNumber(ncono, 2))
        Console.ReadLine()
    End Sub

End Module
```

17 Estudio estadístico (grado de dificultad: 3)

17.1Enunciado

Diseñar un programa en VB en modo Console que permita realizar un estudio estadístico sobre la intención de voto de las próximas elecciones. Para ello, se introducirán para cada población el número de votantes con intención de voto para cada uno de los partidos electorales. Los partidos electorales son: Partido X, Partido Y y Partido Z.

Se construirá una estructura para almacenar los datos de cada provincia que contenga la siguiente información:

Structure datos_estadisticos1
Dim Provincia as String
Dim Partido_x as integer
Dim Partido_y as integer
Dim Partido_z as integer
End structure

Para ello, se creará un vector con el número de poblaciones que se desea realizar el cálculo, máximo 10. Se pretende:

1. Indicar el nombre del partido que ganaría en todas las poblaciones.
2. Indicar para cada población cuál es el partido ganador.
3. Visualizar todos los datos introducidos de todas poblaciones.
4. Salir. Aparezca vuestro nombre y grupo.

Para ello, se creará el siguiente menú:

1. Introducir datos. Mediante un procedimiento.
2. Indicar el nombre del que ganaría en todas las poblaciones. Mediante una función que devuelva un String
3. Indicar el partido ganador en cada una de las poblaciones. Mediante un procedimiento o función.
4. Visualizar todos los datos introducidos de todas poblaciones. Crear una función que me devuelve en un String datos introducidos.
5. Salir. Aparezca vuestro nombre y grupo.

17.2Solución

```
Module Module1
    'Crea estructura de datos
    Structure datos_estadisticos1
        Dim poblacion As String
        Dim Partido_x As Integer
        Dim Partido_y As Integer
        Dim partido_z As Integer
        Dim ganador As String
    End Structure
    Sub Pedir_datos(ByRef datos_provincia() As datos_estadisticos1)
        'pedir datos y calcula provincias mayor y el ganador total
        Dim i, numero As Byte
        Dim total_X As Integer = 0
        Dim total_y As Integer = 0
        Dim total_z As Integer = 0
        Dim mayor As Integer
        Dim ganador As String
        'Pedir cuantas provincias se van a introducir datos
```

```
    Do
        numero = InputBox("cuantos datos provincia")
    Loop Until numero > 0 And numero < 11
    ReDim datos_provincia(numero)
    For i = 0 To UBound(datos_provincia) - 1
        With datos_provincia(i)
            .poblacion = InputBox("dame el nombre :")
            Do
                .Partido_x = InputBox(" partido X")
                If .Partido_x <= 0 Then
                    MsgBox("Debe ser mayor que 0")
                End If
            Loop Until .Partido_x > 0
            mayor = .Partido_x
            ganador = "Partido X"
            Do
                .Partido_y = (InputBox("parido Z"))
                If .Partido_y <= 0 Then
                    MsgBox("Debe ser mayor que 0")
                End If
            Loop Until .Partido_y > 0
            If .Partido_y > mayor Then
                mayor = .Partido_x
                ganador = "Partido Y"
            End If
            Do
                .partido_z = InputBox("partido Z")
                If .partido_z <= 0 Then
                    MsgBox("Debe ser mayor que 0")
                End If
            Loop Until .partido_z > 0
            If .partido_z > mayor Then
                mayor = .partido_z
                ganador = "Partido Z"
            End If
            'Se almacena los datos de cada cada partido por provincia???
            total_X += .Partido_x
            total_y += .Partido_y
            total_z += .partido_z
            'Se almacena el partido ganador por provincia
            .ganador = ganador
        End With

    Next
    'Se almacena por cada provincia el total
    datos_provincia(i).Partido_x = total_X
    datos_provincia(i).Partido_y = total_y
    datos_provincia(i).partido_z = total_z
    'Se localiza cuál es el partido ganador
    If total_X > total_y Then
        mayor = total_X
        ganador = "Partido X"
    Else
        mayor = total_y
        ganador = "Partido Y"
    End If
    If total_z > mayor Then
        ganador = "Partido Z"
    End If
    ' Se almacena el partido ganador
    datos_provincia(i).ganador = ganador
End Sub

Function Imprimir_datos(ByVal datos_provincia() As datos_estadisticos1) As String
```

```
        ' Imprime los datos de todas las provincia
        Dim resultado As String = ""
        Dim datos_provincia_temporal() As datos_estadisticos1
        datos_provincia_temporal = datos_provincia
        Dim total As Long = 0
        Dim i As Byte
        'opcion 3:Visualizar todos los datos introducidos (Partido_x y Partido_y por cada
satos_personal) utilizando un string.
        resultado = "Empresa XX" & vbLf & " Provincia" & vbTab & " Partido X" & vbTab & "
Partido Y" & " Partido Z" & vbLf
        For i = 0 To UBound(datos_provincia) - 1
            With datos_provincia(i)
                resultado = resultado & .poblacion & vbTab & "    " & .Partido_x & vbTab & "
" & .Partido_y & "    " & vbTab & .partido_z & vbLf
            End With
        Next

        Return resultado
    End Function
    Function Calcular_Partido_ganador(ByVal datos_provincia() As datos_estadisticos1) As
String
        'Visualizar partido ganador, el cálculo se realizo al introducir los datos
        Dim ganador_p As String
        ganador_p = datos_provincia(UBound(datos_provincia)).ganador
        Return ganador_p
    End Function
    Function Calcular_Partido_ganador_porprovincia(ByVal datos_provincia() As
datos_estadisticos1) As String
        'devuelve el valor del partido ganador por provincia el cálculo se realizó al pedir
datos
        Dim partido_provincia As String = "El partido por provincia son: "
        For i = 0 To UBound(datos_provincia) - 1
            partido_provincia = partido_provincia & vbLf & datos_provincia(i).poblacion &
vbTab & datos_provincia(i).ganador
        Next
        Return partido_provincia
    End Function
    Sub main()
        'Declaración de variables
        Dim datos_provincia() As datos_estadisticos1
        Dim opcion As Byte
        'Crear el menú
        Do
            'Pedir opción
            opcion = InputBox("Elige la opción:" & vbLf & "1. Introducir datos por
provincia" & vbLf & "2. Visualizar los datos introducidos" & vbLf & "3.Visualizar los
ganadores por provincias" & vbLf & "4.Visualizar el partido ganador" & vbLf & "5.Salir.")
            Select Case opcion
                Case 1
                    'Opción 1, pedir datos
                    MsgBox("Pedir datos")

                    Pedir_datos(datos_provincia)
                Case 2
                    'Opción 2:  Visualizar todos los datos introducidos (Partido_x y
Partido_y por cada satos_personal) utilizando un string.
                    If IsNothing(datos_provincia) Then
                        MsgBox("Debes introducir primero los datos")
                    Else
                        MsgBox(Imprimir_datos(datos_provincia))
                    End If

                Case 3
                    'opcion 3:Visualizar los ganadores por provincia.
```

```
                    If IsNothing(datos_provincia) Then
                        MsgBox("Debes introducir primero los datos")
                    Else
                        MsgBox(Calcular_Partido_ganador_porprovincia(datos_provincia))
                    End If
                Case 4
                    'opcion 4: Visualizar partido ganador
                    If IsNothing(datos_provincia) Then
                        MsgBox("Debes introducir primero los datos")
                    Else
                        MsgBox("Partido ganador====>" &
Calcular_Partido_ganador(datos_provincia))
                    End If
                Case 5
                    ' opcion 5 Salir
                    MsgBox("Salir. Programa realizado por Juan Penas Penas")
                Case Else
                    'Teclee u número de opción incorrecto
                    MsgBox("No pusiste opción correcta del menú")
            End Select
        Loop Until opcion = 5
    End Sub
End Module
```

18 Cafetería (grado de dificultad: 3)

18.1Enunciado

Diseñar un programa en VB en modo console que permita gestionar los ingresos de la cafetería de a la Escuela de Industriales en una determinada semana. Se trata de almacenar los ingresos de 5 días hábiles (lunes-viernes) y dos turnos (Mañana y Tarde). Se introducirán las ganancias de cada uno de los días y turnos, mediante una función que me devuelva el vector con todos los ingresos que se han generado a lo largo de la semana. Para ello, se deberá construir una estructura para almacenar la información:

Structure semana
Dim manana As Single
Dim tarde As Single
End Structure

Una vez introducidos los datos, se mostrará el siguiente menú:

1. Indicar que día (Lunes…Viernes) ha tenido el mayor ingreso y su cantidad. Utilizando una función una función que devuelva en un string los días con mayores ventas.
2. Visualizar el promedio diario (en un string), utilizando un procedimiento que calcule el promedio diario
3. Visualizar los ingresos de todos los días para todos los turnos (en un string). Utilizando una función que me devuelva lo que ha se ha generado a lo largo de la semana.
4. Salir

18.2Solución

```
Module Module1
    'Crear la estructura
    Structure semana
        Dim manana As Single
        Dim tarde As Single
        Dim total As Double
    End Structure
    Function Pedir_datos(ByVal dia() As String) As semana()
        'Procedimiento que permite introducir las entradas de la cafetería para cada uno de
los días
        Dim i As Integer
        Dim cafeteria(4) As semana
        For i = 0 To 4
            With cafeteria(i)
                Do
                    .manana = InputBox("dame ganancia día: " & dia(i) & " turno Mañana")
                    If .manana < 0 Then
                        MsgBox("error en los dataos")
                    End If
                Loop Until .manana >= 0
                .tarde = InputBox("dame ganancia día: " & dia(i) & " turno Tarde ")
                If .tarde < 0 Then
                    MsgBox("error en los dtaos")
                End If
                'ALmacenamos las ganancias diarias en la fila 2
                .total = .manana + .tarde
            End With
        Next
        Return cafeteria
    End Function
    Function Calcular_repetidos(ByVal dia() As String, ByVal cafeteria() As semana) As
String
```

```
        'Calcula si hay más de dos días con las mismas ganancias
        Dim existe As Integer
        Dim i, fila_valor, mayor As Byte
        Dim repetidos As String
        mayor = cafeteria(0).total
        fila_valor = 0
        existe = 0
        repetidos = ""
        For i = 1 To 4
            If cafeteria(i).total >= mayor Then
                mayor = cafeteria(i).total
                fila_valor = i
                existe = existe + 1
            End If
        Next
        If existe = 1 Then
            repetidos = "el dia con más ganancias: " & dia(fila_valor) & " número ganancias:
" & cafeteria(fila_valor).total
        Else
            For i = 0 To 4
                If mayor = cafeteria(i).total Then
                    fila_valor = i
                    repetidos = repetidos & dia(fila_valor) & " numero ganancias: " &
cafeteria(fila_valor).total & vbLf
                End If
            Next
            repetidos = "los dias con más ganancias: " & vbLf & repetidos
        End If
        Return repetidos
    End Function
    Function Imprimir_datos(ByVal dia() As String, cafeteria() As semana) As String
        'procedimiento para imprimir los datos de la semana
        Dim cafeteria_resultado As String
        Dim i As Integer
        cafeteria_resultado = "          Mañana       Tarde       Total" & vbLf
        For i = 0 To 4
            With cafeteria(i)
                cafeteria_resultado = cafeteria_resultado & dia(i) & vbTab & .manana & vbTab
& .tarde & vbTab & .total & vbLf
            End With
        Next
        Return cafeteria_resultado
    End Function
    Sub calculos_promedio(ByVal dia() As String, ByVal cafeteria() As semana)
        'Procedimiento para imprimir el promedio
        Dim i As Byte
        Dim promedio As Single
        Dim resultado_promedio, cafeteria_resultado As String
        resultado_promedio = ""
        cafeteria_resultado = ""
        promedio = 0
        For i = 0 To 4
            promedio = Math.Round(cafeteria(i).total / 2, 1)
            cafeteria_resultado = cafeteria_resultado & dia(i) & vbTab & promedio & vbLf
        Next
        MsgBox(cafeteria_resultado, MsgBoxStyle.Information, "cafeteria de promedios")
    End Sub
    Sub Main()
        'Declaración de variables
        Dim cafeteria(4) As semana
        Dim dia() As String = {"Lunes", "Martes", "Miércoles", "Jueves", "viernes"}
        Dim horarios() As String = {"manana ", "Tarde"}
        Dim cafeteria_resultado As String = ""
        Dim repetidos As String = ""
```

```
        Dim cafeteria_promedio As String = ""
        Dim opcion As Byte
        'Pedir los datos de las ventas semanlmente
        cafeteria = Pedir_datos(dia)
        'Creamos el menú y las diferentes opciones
        Do
            opcion = InputBox("Elige la opción:" & vbLf & "1. Indicar que día ha tenido el
mayor ingreso y su cantidad" & vbLf & "2. Visualizar el promedio diario" & vbLf &
"3.Visualizar los ingresos de todos los días para todos los turnos: Total" & vbLf &
"4.Salir.")
            Select Case opcion
                Case 1
                    ' dia mas ganancia
                    MsgBox(Calcular_repetidos(dia, cafeteria))
                Case 2
                    'Calculo del promedio diario
                    calculos_promedio(dia, cafeteria)
                Case 3
                    'Imprime las ventas de una semana
                    MsgBox(Imprimir_datos(dia, cafeteria))
                Case 4
                    'Salir del pragrama
                    MsgBox("Salir  ")
                Case Else
                    'Si no tecleo un número correcto  del menú
                    MsgBox("no pusiste algo correcto")
            End Select
        Loop Until opcion = 4
    End Sub
End Module
```

19 Programa que permite introducir notas de alumnos (evolución I) (grado de dificultad: 3)

19.1Enunciado

El profesor de la asignatura “Informática aplicada a la ingeniería” desea crear un programa que le permita introducir las notas de los alumnos (máximo 50) de todo el cuatrimestre

Teniendo en cuenta:

1. Debe introducir el número de alumnos.
2. Todos los alumnos son españoles (tienen NIF).
3. El alumno que realiza evaluación continua, realizará 4 exámenes y su nota final es la media de las 4 pruebas realizadas. Si el alumno renuncia a la evaluación continua, su nota final será la que obtenga en el examen de mayo.
4. Una vez que se hayan introduciendo las datos. Habrá un menú, en donde el profesor decidirá modificar una nota de un alumno y/o ver un listado de alumnos.

19.2Solución

```
Module Module1
  'Declaración de una nueva variable llamada alumno
  Structure alumno
    Public DNI As String
    Public nombre As String
    Public Evaluacion_continua As Boolean
    Public prueba1 As Double
    Public prueba2 As Double
    Public prueba3 As Double
    Public prueba4 As Double
    Public media As Double
  End Structure
  'Pulsar cualquier tecla para continuar
  Sub espera_intro(ByRef mensaje As String)
    Console.WriteLine(mensaje)
    Console.ReadLine()
  End Sub
  'Deja dos líneas en blanco
  Sub saltos()
    Console.WriteLine()
    Console.WriteLine()
  End Sub
  ' Borrar pantalla
  Sub limpiar()
    Console.Clear()
  End Sub
  ' Función permite comprobar que la nota introducuida sea número
  Function Comprobar_si_es_prueba(ByVal prueba As String) As Boolean
    If Val(prueba) >= 0 And Val(prueba) <= 10 Then
      Comprobar_si_es_prueba = True
    Else : Comprobar_si_es_prueba = False
      Console.WriteLine("La nota debe ser numérica y/o menor o igual que 10,
vuelve a introducirla")
    End If
  End Function
  ' Un procedimiento para pedir el número de alumnos y comprobar que no sea
mayor que 50
```

```
  Sub numero_alumnos(ByRef tope As Byte)
    Do
      Console.Write("Dame el número total de alumnos a introducir: ")
      tope = Console.ReadLine()
    Loop Until tope <= 50
  End Sub
  'Función que permite verificar si el DNI contiene 8 números y una letra al
final
  Function comprobar_DNI(ByVal temp As String) As Boolean
    Dim Numero_caracteres As Integer
    Dim numero As String
    Dim letras As Char
    Numero_caracteres = Len(temp)
    numero = Mid(temp, 1, 8)
    letras = UCase(Mid(temp, 9, 9))
    If Numero_caracteres = 9 And IsNumeric(numero) And letras Like "[A-Z]" Then
      comprobar_DNI = True
    Else
      Console.WriteLine("DNI Incorrecto, debe contener 8 dígitos y una letra")
      comprobar_DNI = False
    End If
  End Function
  'Función que permite comrpobar que el nombre contenga solamente letras
  Function comprobar_nombre(ByVal temp As String) As Boolean
    Dim Numero_caracteres, i, numero_caracteres_c As Integer
    Dim letras As Char
    Numero_caracteres = Len(temp)
    letras = UCase(Mid(temp, 1, Numero_caracteres))
    i = 1
    numero_caracteres_c = 0
    Do
      letras = UCase(Mid(temp, i, i))
      If letras Like "[A-Z]" Or letras = Chr(32) Then
        numero_caracteres_c = numero_caracteres_c + 1
      End If
      i = i + 1
    Loop Until i = Numero_caracteres + 1
    If numero_caracteres_c = Numero_caracteres Then
      comprobar_nombre = True
    Else
      comprobar_nombre = False
    End If
  End Function
  'Pedir los datos de un alumno y verificarlos
  Sub pedir_datos(ByRef temporal As alumno, ByVal i As Byte, ByVal matriz() As
alumno)
    Dim continua As Char
    Dim Temp As String
    Dim cierto As Boolean
    Dim posicion As Byte
    With temporal
      cierto = False
      Do
        Console.Write("Dame el DNI del alumno ===>")
        Temp = Console.ReadLine()
        Temp = UCase(Trim(Temp))
        If i = 0 Then
          cierto = False
        Else
          cierto = DNI_repetido(matriz, i, Temp, posicion)
          If DNI_repetido(matriz, i, Temp, posicion) = True Then
            saltos()
            Console.WriteLine("Este alumno ya existe")
```

```
          End If
        End If
      Loop Until comprobar_DNI(Temp) = True And cierto = False
      .DNI = Temp
      Do
        Console.Write("Dame el Nombre del alumno ===>")
        Temp = Console.ReadLine()
      Loop Until comprobar_nombre(Temp) = True
      .nombre = Temp
      .Evaluacion_continua = True
      Console.Write("Renuncia a la evaluación continua (Teclee S, cualquier otro
caracter es evaluación continua): ")
      continua = Console.ReadLine()
      If UCase(continua) = "S" Then
        .Evaluacion_continua = False
      End If
      If .Evaluacion_continua = True Then
        Do
          Console.Write("Dame la nota del primer examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba1 = Val(Temp)
        Do
          Console.Write("Dame la nota del segundo examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba2 = Val(Temp)
        Do
          Console.Write("Dame la nota del tercer examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba3 = Val(Temp)
        Do
          Console.Write("Dame la nota del último examen (Fecha Oficial Mayo): ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba4 = Val(Temp)
      End If
      If .Evaluacion_continua = False Then
        Do
          Console.Write("Dame la Nota del examen (Fecha Oficial Mayo): ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba1 = 0
        .prueba2 = 0
        .prueba3 = 0
        .prueba4 = 0
        .media = Val(Temp)
      Else
        If (.prueba1 + .prueba2 + .prueba3 + .prueba4) = 0 Then
          .media = 0
        Else
          .media = Math.Round((.prueba1 + .prueba2 + .prueba3 + .prueba4) / 4,
2)
        End If
      End If
    End With
  End Sub
  'Verificar que un NIF no este dupicado o exista un alumno con ese NIF
  Function DNI_repetido(ByRef matriz() As alumno, ByVal tope As Byte, ByVal temp
As String, ByRef posicion As Byte) As Boolean
    Dim i As Byte = 0
    DNI_repetido = False
```

```
    Do
      If matriz(i).DNI = temp Then
        DNI_repetido = True
        posicion = i
        i = tope
      End If
      i = i + 1
    Loop Until i = tope + 1
  End Function
  'Permite modificar una nota de un alumno introduciendo el NIF
  Sub modifica_calificacion(ByRef matriz() As alumno, ByVal tope As Byte)
    Dim temp As String
    Dim i As Byte = 0
    Dim posicion As Byte
    Dim convocatoria As Byte
    Console.Write("Dame el NIF del alumno ===>")
    temp = Console.ReadLine()
    temp = UCase(Trim(temp))
    If DNI_repetido(matriz, tope, temp, posicion) = False Then
      Console.Write("Ese alumno con ese DNI no existe")
    Else
      If matriz(posicion).Evaluacion_continua = False Then
        Console.Write("Del alumno: " & matriz(posicion).nombre)
        Do
          Console.Write(" Dame la Nota del examen (Fecha Oficial Mayo): ")
          temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(temp) = True
        matriz(posicion).media = Val(temp)
      Else
        Do
          Console.Write("Del alumno: " & matriz(posicion).nombre & " , dame la
nueva nota: ")
          temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(temp) = True
        Do
          Console.Write("De que convocatoria vas a cambiar la nota ")
          convocatoria = Console.ReadLine()
        Loop Until Val(convocatoria) <= 4 And Val(convocatoria) >= 1
        Console.WriteLine("convocatoria" & convocatoria)
        convocatoria = convocatoria - 1
        Console.WriteLine("convocatoria 2" & convocatoria)
        Select Case convocatoria
          Case 0
            matriz(posicion).prueba1 = Val(temp)
          Case 1
            matriz(posicion).prueba2 = Val(temp)
          Case 2
            matriz(posicion).prueba3 = Val(temp)
          Case 3
            matriz(posicion).prueba4 = Val(temp)
        End Select
        matriz(posicion).media = (matriz(posicion).prueba1 +
matriz(posicion).prueba2 + matriz(posicion).prueba3 + matriz(posicion).prueba4)
/ 4
      End If
    End If
  End Sub
  ' Permite imprimir todas las calificaciones de los alumnos (genera u listado).
  Sub Imprimir_datos(ByVal matriz() As alumno, ByVal tope As Byte)
    Dim temporal As alumno
    Dim i As Byte
    Dim imprimir As String
```

```
    Console.WriteLine("NIF " & vbTab & vbTab & "Nombre " & vbTab & vbTab & "Nota
1" & vbTab & "Nota 2" & vbTab & "Nota 3" & vbTab & "Nota 4" & vbTab & "Media")
    For i = 0 To tope - 1
      temporal = matriz(i)
      With temporal
        If temporal.Evaluacion_continua = True Then
          imprimir = .DNI & vbTab & .nombre & vbTab & vbTab & .prueba1 & vbTab &
.prueba2 & vbTab & .prueba3 & vbTab & .prueba4 & vbTab & .media
        Else
          imprimir = .DNI & vbTab & .nombre & vbTab & vbTab & " " & vbTab & " "
& vbTab & " " & vbTab & " " & vbTab & .media
        End If
        Console.Write(imprimir)
      End With
      Console.WriteLine()
    Next
  End Sub
  Sub menu()
    saltos()
    Console.WriteLine("Menú")
    Console.WriteLine("1. Modificar una calificación")
    Console.WriteLine("2. Listar")
    Console.WriteLine("3. Salir")
    Console.Write("Cuál es tu opción ====> ")
  End Sub
  Sub Main()
    'Declaración de variables
    Dim tope, i As Byte
    Dim temporal As alumno
    Dim matriz() As alumno
    Dim opcion As Byte
    'Pedir cuantos alumno hay y verificar que sea inferior o igual a 50
    numero_alumnos(tope)
    'Redimensionar el arreglo con el número de alumnos
    ReDim matriz(tope)
    'Pedir nombre y caificaciones de los alumnos
    For i = 0 To tope - 1
      pedir_datos(temporal, i, matriz)
      matriz(i) = temporal
    Next
    ' Imprimir todas las notas de todos los alumnos
    saltos()
    Console.WriteLine("LISTADO DE ALUMNOS INTRODUCIDOS")
    saltos()
    Imprimir_datos(matriz, tope)
    saltos()
    espera_intro("teclea <Intro> para continuar")
    limpiar()
    'Crear un menú que permite modificar una nota, imprimir los alumnos o salir
del program
    Do
      menu()
      opcion = Console.ReadLine()
      Select Case opcion
        Case 1
          Console.WriteLine("Modificar calificacion")
          modifica_calificacion(matriz, tope)
        Case 2
          limpiar()
          Console.WriteLine("LISTADO DE ALUMNOS")
          Imprimir_datos(matriz, tope)
      End Select
    Loop Until opcion = 3
```

```
    limpiar()
    espera_intro("teclea <Intro> para salir")
  End Sub
End Module
```

20 Programa para introducir las notas de los alumnos (evolución II) (grado de dificultad: 3)

20.1 Enunciado

El profesor de la asignatura “Informática aplicada a la ingeniería” desea crear un programa que le permita introducir las notas de los alumnos (máximo 50) de todo el cuatrimestre, sin utilizar estructuras.

Teniendo en cuenta:

1. Debe introducir el número de alumnos.
2. Todos los alumnos son españoles (tienen NIF).
3. El alumno que realiza evaluación continua, realizará 4 exámenes y su nota final es la media de las 4 pruebas realizadas. Si el alumno renuncia a la evaluación continua, su nota final será la que obtenga en el examen de mayo.
4. Una vez que se hayan introduciendo las datos. Habrá un menú, en donde el profesor decidirá modificar una nota de un alumno y/o ver un listado de alumnos.

20.2 Solución

```
Module Module1
  'Pulsar cualquier tecla para continuar
  Sub espera_intro(ByRef mensaje As String)
    Console.WriteLine(mensaje)
    Console.ReadLine()
  End Sub
  'Deja dos líneas en blanco
  Sub saltos()
    Console.WriteLine()
    Console.WriteLine()
  End Sub
  ' Borrar pantalla
  Sub limpiar()
    Console.Clear()
  End Sub
  ' Función permite comprobar que la nota introducuida sea número
  Function Comprobar_si_es_prueba(ByVal prueba As String) As Boolean
    If Val(prueba) >= 0 And Val(prueba) <= 10 Then
      Comprobar_si_es_prueba = True
    Else : Comprobar_si_es_prueba = False
      Console.WriteLine("La nota debe ser numérica y/o menor o igual que 10,
vuelve a introducirla")
    End If
  End Function
  ' Un procedimiento para pedir el número de alumnos y comprobar que no sea
mayor que 50
  Sub numero_alumnos(ByRef tope As Byte)
    Do
      Console.Write("Dame el número total de alumnos a introducir: ")
      tope = Console.ReadLine()
    Loop Until tope <= 50
  End Sub
  'Función que permite verificar si el DNI contiene 8 números y una letra al
final
  Function comprobar_DNI(ByVal temp As String) As Boolean
    Dim Numero_caracteres As Integer
    Dim numero As String
```

```
    Dim letras As Char
    Numero_caracteres = Len(temp)
    numero = Mid(temp, 1, 8)
    letras = UCase(Mid(temp, 9, 9))
    If Numero_caracteres = 9 And IsNumeric(numero) And letras Like "[A-Z]" Then
      comprobar_DNI = True
    Else
      Console.WriteLine("DNI Incorrecto, debe contener 8 dígitos y una letra")
      comprobar_DNI = False
    End If
  End Function
  'Función que permite comrpobar que el nombre contenga solamente letras
  Function comprobar_nombre(ByVal temp As String) As Boolean
    Dim Numero_caracteres, i, numero_caracteres_c As Integer
    Dim letras As Char
    Numero_caracteres = Len(temp)
    letras = UCase(Mid(temp, 1, Numero_caracteres))
    i = 1
    numero_caracteres_c = 0
    Do
      letras = UCase(Mid(temp, i, i))
      If letras Like "[A-Z]" Or letras = Chr(32) Then
        numero_caracteres_c = numero_caracteres_c + 1
      End If
      i = i + 1
    Loop Until i = Numero_caracteres + 1
    If numero_caracteres_c = Numero_caracteres Then
      comprobar_nombre = True
    Else
      comprobar_nombre = False
    End If
  End Function
  'Pedir los datos de un alumno y verificarlos
  Sub pedir_datos(ByVal i As Byte, ByRef matriz(,) As String)
    Dim continua As Char
    Dim Temp As String
    Dim cierto As Boolean
    Dim posicion As Byte
    cierto = False
    Do
      Console.Write("Dame el DNI del alumno ===>")
      Temp = Console.ReadLine()
      Temp = UCase(Trim(Temp))
      If i = 0 Then
        cierto = False
      Else
        cierto = DNI_repetido(matriz, i, Temp, posicion)
        If DNI_repetido(matriz, i, Temp, posicion) = True Then
          saltos()
          Console.WriteLine("Este alumno ya existe")
        End If
      End If
    Loop Until comprobar_DNI(Temp) = True And cierto = False
    matriz(i, 0) = Temp
    Do
      Console.Write("Dame el Nombre del alumno ===>")
      Temp = Console.ReadLine()
    Loop Until comprobar_nombre(Temp) = True
    matriz(i, 1) = Temp
    matriz(i, 2) = True
    Console.Write("Renuncia a la evaluación continua (Teclee S, cualquier otro
caracter es evaluación continua): ")
    continua = Console.ReadLine()
```

```
    If UCase(continua) = "S" Then
      matriz(i, 2) = False
    End If
    If matriz(i, 2) = True Then
      Do
        Console.Write("Dame la nota del primer examen: ")
        Temp = Console.ReadLine()
      Loop Until Comprobar_si_es_prueba(Temp) = True
      matriz(i, 3) = Val(Temp)
      Do
        Console.Write("Dame la nota del segundo examen: ")
        Temp = Console.ReadLine()
      Loop Until Comprobar_si_es_prueba(Temp) = True
      matriz(i, 4) = Val(Temp)
      Do
        Console.Write("Dame la nota del tercer examen: ")
        Temp = Console.ReadLine()
      Loop Until Comprobar_si_es_prueba(Temp) = True
      matriz(i, 5) = Val(Temp)
      Do
        Console.Write("Dame la nota del último examen (Fecha Oficial Mayo): ")
        Temp = Console.ReadLine()
      Loop Until Comprobar_si_es_prueba(Temp) = True
      matriz(i, 6) = Val(Temp)
    End If
    If matriz(i, 2) = False Then
      Do
        Console.Write("Dame la Nota del examen (Fecha Oficial Mayo): ")
        Temp = Console.ReadLine()
      Loop Until Comprobar_si_es_prueba(Temp) = True
      matriz(i, 3) = 0
      matriz(i, 4) = 0
      matriz(i, 5) = 0
      matriz(i, 6) = 0
      matriz(i, 7) = Val(Temp)
    Else
      If (Val(matriz(i, 3)) + Val(matriz(i, 4)) + Val(matriz(i, 5)) +
Val(matriz(i, 6))) = 0 Then
        matriz(i, 7) = 0
      Else
        matriz(i, 7) = Math.Round((Val(matriz(i, 3)) + Val(matriz(i, 4)) +
Val(matriz(i, 5)) + Val(matriz(i, 6))) / 4, 2)
      End If
    End If
  End Sub
  'Verificar que un NIF no este dupicado o exista un alumno con ese NIF
  Function DNI_repetido(ByRef matriz(,) As String, ByVal tope As Byte, ByVal
temp As String, ByRef posicion As Byte) As Boolean
    Dim i As Byte = 0
    DNI_repetido = False
    Do
      If matriz(i, 0) = temp Then
        DNI_repetido = True
        posicion = i
        i = tope
      End If
      i = i + 1
    Loop Until i = tope + 1
  End Function
  'Permite modificar una nota de un alumno introduciendo el NIF
  Sub modifica_calificacion(ByRef matriz(,) As String, ByVal tope As Byte)
    Dim temp As String
    Dim i As Byte = 0
```

```
    Dim posicion As Byte
    Dim convocatoria As Byte
    Console.Write("Dame el NIF del alumno ===>")
    temp = Console.ReadLine()
    temp = UCase(Trim(temp))
    If DNI_repetido(matriz, tope, temp, posicion) = False Then
      Console.Write("Ese alumno con ese DNI no existe")
    Else
      If matriz(posicion, 2) = False Then
        Console.Write("Del alumno: " & matriz(posicion, 1))
        Do
          Console.Write(" Dame la Nota del examen (Fecha Oficial Mayo): ")
          temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(temp) = True
        matriz(posicion, 7) = Val(temp)
      Else
        Do
          Console.Write("Del alumno: " & matriz(posicion, 1) & " , dame la nueva
nota: ")
          temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(temp) = True
        Do
          Console.Write("De que convocatoria vas a cambiar la nota ")
          convocatoria = Console.ReadLine()
        Loop Until Val(convocatoria) <= 4 And Val(convocatoria) >= 1
        Console.WriteLine("convocatoria" & convocatoria)
        convocatoria = convocatoria - 1
        Console.WriteLine("convocatoria 2" & convocatoria)
        Select Case convocatoria
          Case 0
            matriz(posicion, 3) = Val(temp)
          Case 1
            matriz(posicion, 4) = Val(temp)
          Case 2
            matriz(posicion, 5) = Val(temp)
          Case 3
            matriz(posicion, 6) = Val(temp)
        End Select
        matriz(posicion, 7) = (matriz(posicion, 3) + matriz(posicion, 4) +
matriz(posicion, 5) + matriz(posicion, 6)) / 4
      End If
    End If
  End Sub
  ' Permite imprimir todas las calificaciones de los alumnos (genera u listado).
  Sub Imprimir_datos(ByVal matriz(,) As String, ByVal tope As Byte)
    Dim i As Byte
    Dim imprimir As String
    Console.WriteLine("NIF " & vbTab & vbTab & "Nombre " & vbTab & vbTab & "Nota
1" & vbTab & "Nota 2" & vbTab & "Nota 3" & vbTab & "Nota 4" & vbTab & "Media")
    For i = 0 To tope - 1
      If matriz(i, 2) = True Then
        imprimir = matriz(i, 0) & vbTab & matriz(i, 1) & vbTab & vbTab &
matriz(i, 3) & vbTab & matriz(i, 4) & vbTab & matriz(i, 5) & vbTab & matriz(i,
6) & vbTab & matriz(i, 7)
      Else
        imprimir = matriz(i, 0) & vbTab & matriz(i, 1) & vbTab & vbTab & " " &
vbTab & " " & vbTab & " " & vbTab & " " & vbTab & matriz(i, 7)
      End If
      Console.Write(imprimir)
      Console.WriteLine()
    Next
  End Sub
  Sub menu()
```

```
    saltos()
    Console.WriteLine("Menú")
    Console.WriteLine("1. Modificar una calificación")
    Console.WriteLine("2. Listar")
    Console.WriteLine("3. Salir")
    Console.Write("Cuál es tu opción ====> ")
  End Sub
  Sub Main()
    'Declaración de variables
    Dim tope, i As Byte
    Dim temporal(,) As String
    Dim matriz(,) As String
    Dim opcion As Byte
    'Pedir cuantos alumno hay y verificar que sea inferior o igual a 50
    numero_alumnos(tope)
    'Redimensionar el arreglo con el número de alumnos
    ReDim matriz(tope, 7)
    'Pedir nombre y caificaciones de los alumnos
    For i = 0 To tope - 1
      pedir_datos(i, matriz)
    Next
    ' Imprimir todas las notas de todos los alumnos
    saltos()
    Console.WriteLine("LISTADO DE ALUMNOS INTRODUCIDOS")
    saltos()
    Imprimir_datos(matriz, tope)
    saltos()
    espera_intro("teclea <Intro> para continuar")
    limpiar()
    'Crear un menú que permite modificar una nota, imprimir los alumnos o salir
del program
    Do
      menu()
      opcion = Console.ReadLine()
      Select Case opcion
        Case 1
          Console.WriteLine("Modificar calificacion")
          modifica_calificacion(matriz, tope)
        Case 2
          limpiar()
          Console.WriteLine("LISTADO DE ALUMNOS")
          Imprimir_datos(matriz, tope)
      End Select
    Loop Until opcion = 3
    limpiar()
    espera_intro("teclea <Intro> para salir")
  End Sub
End Module
```

21 Multiplicar dos vectores de igual longitud y guardar el resultado en una matriz (grado de dificultad: 3)

21.1Enunciado

Aparecerá un menú con las opciones 1. Rellenar Vectores 2. Imprimir multiplicación 3. Rellenar/Imprimir Matriz 4. Salir. El programa debe repetirse mientras el usuario no seleccione la opción 4. En caso de escoger otra opción debe aparecer un mensaje de error.

1. En la opción 1, se rellenarán dos vectores de 12 elementos con números aleatorios del 10-1000 utilizando la función random. CInt((MAXIMO – MINIMO+1) * Rnd() + MINIMO).
2. Se muestran ambos vectores en pantalla.
3. En la opción 2, se multiplica el elemento de la primera posición de un vector con el elemento de la última posición del otro vector, y así hasta multiplicar todos los elementos de ambos vectores.
4. Rellenar una matriz de 3X4 con el resultado de la multiplicación de los vectores anteriores.
5. Imprimir la matriz.

21.2Solución sin Funciones

```
'Declaración de las constantes y variables que se utilizarán en el main
        Const numeroElementos As Integer = 12 'cantidad de elementos de ambos vectores
        Const MAXIMO As Integer = 1000 'valor aleatorio máximo para generar
        Const MINIMO As Integer = 10 'valor aleatorio mínimo para generar
        Dim posic As Integer = 0
        Dim vector(numeroElementos - 1) As Integer ' Creación de los 3 vectores necesarios
en el problema y que tienen la misma dimensión.
        Dim vector1(numeroElementos - 1) As Integer
        Dim vector2(numeroElementos - 1) As Integer
        Dim mensaje As String = "" 'variables que se utilizan para concatener los elementos
de cada uno de los vectores
        Dim mensaje1 As String = ""
        Dim mensaje2 As String = ""
        Dim opcion As Char 'lee la opción del usuario
        Dim matriz(3 - 1, 4 - 1) As Integer 'Declara y dimesiona la matriz de 3*4
        Randomize() 'Generar aleatorios
        Do
            opcion = InputBox("R. Rellenar vectores" & vbNewLine & "M. Multiplicar" &
vbNewLine & "I. Rellenar/Imprimir Matriz" & vbNewLine & "S. Salir" & vbNewLine & "Opción")
            opcion = UCase(opcion) ' convertir a mayúscula el caracter leído
            Select Case opcion
                Case "R"
                    mensaje = "" ' Inicializar el String para que no contenga restos de
ejecuciones anteriores
                    mensaje1 = ""
                    For i = 0 To numeroElementos - 1 'ciclo para rellenar al mismo tiempo
los dos vectores desde la posición 0 a la 11 (12-1)
                        vector(i) = CInt((MAXIMO - MINIMO + 1) * Rnd() + MINIMO) 'Crea un
número aleatorio en el rango establecido para el vector
                        vector1(i) = CInt((MAXIMO - MINIMO + 1) * Rnd() + MINIMO) 'Crea un
número aleatorio en el rango establecido para el vector1
                        mensaje = mensaje & CStr(vector(i)) & "   " ' concatena los
elementos del vector en mensaje
                        mensaje1 = mensaje1 & CStr(vector1(i)) & "   " ' concatena los
elementos del vector1 en mensaje1
                    Next
                Case "M"
```

```
                    For i = 0 To numeroElementos - 1 'Recorre los vectores, el vector desde
la posición 0 hasta 11 y el vector1 desde la posición 11-0
                        vector2(i) = vector(i) * vector1(numeroElementos - 1 - i) 'el
resultado de la multiplicación se almacena en el vector2
                        mensaje2 = mensaje2 & CStr(vector2(i)) & "   " ' el resultado se
almacena convertido en String en la variable mensaje
                    Next
                    MsgBox(mensaje & vbNewLine & mensaje1 & vbNewLine & mensaje2) 'imprime
tres variables separadas por un salto de línea
                Case "I"
                    mensaje = "" 'inicializa la variable mensaje
                    posic = 0 ' inicia en la primera posición válida del vector. Se utiliza
para recorrer el vector
                    'Recorrido de la matriz utilizando dos ciclos uno para el desplazamiento
horizontal y otro para la vertical
                    For i = 0 To UBound(matriz, 1) '  ciclo para recorrer la primera
dimensión de la matriz desde 0 hasta el indice del elemento mayor
                        For j = 0 To UBound(matriz, 2)  ' ciclo para recorrer la segunda
dimensión de la matriz desde 0 hasta el indice del elemento mayor
                            matriz(i, j) = vector2(posic)
                            posic = posic + 1 ' incrementar en uno la posición del vector
                            mensaje = mensaje & CStr(matriz(i, j)) & "   " ' concatena cada
elemento que acaba de almacenarse en la matriz en la variable mensaje
                        Next
                        mensaje = mensaje & vbNewLine ' concatena un salto de líne por cada
desplazamiento horizontal de la matriz
                    Next
                    MsgBox(mensaje)
            End Select
        Loop Until opcion = "S"c 'ciclo para que el usuario seleccione opción mientas no
teclee S(Salir)
    End Sub
End Module
```

21.3Solución con funciones

```
Module Module3
    Sub main()
        Const numeroElementos As Integer = 12 'cantidad de elementos de ambos vectores
        Const MAXIMO As Integer = 1000 'valor aleatorio máximo para generar
        Const MINIMO As Integer = 10 'valor aleatorio mínimo para generar
        Dim posic As Integer = 0
        Dim vector(numeroElementos - 1) As Integer ' Creación de los 3 vectores necesarios
en el problema y que tienen la misma dimensión.
        Dim vector1(numeroElementos - 1) As Integer
        Dim vector2(numeroElementos - 1) As Integer
        Dim opcion As Char
        Dim matriz(3 - 1, 4 - 1) As Integer  'Declara y dimesiona la matriz de 3*4
        Randomize()
        Do
            opcion = InputBox("R. Rellenar vectores" & vbNewLine & "M. Multiplicar" &
vbNewLine & "I. Rellenar/Imprimir Matriz" & vbNewLine & "S. Salir" & vbNewLine & "Opción")
            opcion = UCase(opcion)
            Select Case opcion
                Case "R"
                    RellenarVector(vector, numeroElementos, MAXIMO, MINIMO)
                    RellenarVector(vector1, numeroElementos, MAXIMO, MINIMO)
                    MsgBox(ImprimirVector(vector1, numeroElementos) & vbNewLine &
ImprimirVector(vector1, numeroElementos))
                Case "M"
                    Multiplica(vector, vector1, vector2, numeroElementos)
                    MsgBox(ImprimirVector(vector2, numeroElementos))
                Case "I"
```

```
                    RellenaMatriz(matriz, vector2)
                    MsgBox(ImprimirMatriz(matriz))
                End Select
            Loop Until opcion = "S"c

    End Sub
    'Crear una rutina que recibe como parámetros de entrada:
    'v():vector que recibe
    'num:equivale al número de elemento del vector
    'MAXIMO valor máximo que puede almacenar el vector
    'MINIMO valor mínimo que puede almacenar el vector
    Sub RellenarVector(v() As Integer, num As Integer, MAXIMO As Integer, MINIMO As Integer)
        For i = 0 To num - 1
            v(i) = CInt((MAXIMO - MINIMO + 1) * Rnd() + MINIMO)
        Next
    End Sub
    'Crear función que recibe como parámetro de entrada
    'v():vector que recibe
    'num:equivale al número de elemento del vector
    'Regresa:
    'Un string donde se concatena todos los elementos del vector
    Function ImprimirVector(v() As Integer, num As Integer) As String
        Dim result As String = ""
        For i = 0 To num - 1
            result = result & CStr(v(i)) & "   "
        Next
        ImprimirVector = result
    End Function
    'Procedimiento que recibe como valores de entrada
    'v() y v1():vectores que recibe para multiplicar
    'v2():vector donde se almacena el resultado de la multiplicación
    'num: número de elementos de los tres vectores
    Sub Multiplica(v() As Integer, v1() As Integer, v2() As Integer, num As Integer)
        For i = 0 To num - 1
            v2(i) = v(i) * v1(num - 1 - i)
        Next
    End Sub
    'Procedimientos que recibe como parámetros de entrada
    'm():matriz de 3*4 creada y dimensionada
    'v():vector del cual se copiarán los elementos
    Sub RellenaMatriz(m(,) As Integer, v() As Integer)
        Dim posic As Integer = 0 ' variable que inicia en 0 y se incrementará en 1 en por
cada elemento. Representa el índice de la matriz
        For i = 0 To UBound(m, 1) ' ciclo para recorrer la primera dimensión de la matriz
desde 0 hasta el indice del elemento mayor
            For j = 0 To UBound(m, 2) ' ciclo para recorrer la segunda dimensión de la
matriz desde 0 hasta el indice del elemento mayor
                m(i, j) = v(posic) ' copia el elementos del vector a la matriz
                posic = posic + 1
            Next
        Next
    End Sub
    'Función que recorre la matriz y retorna un string conteniendo todos los elementos
    Function ImprimirMatriz(m(,) As Integer) As String
        Dim mensaje As String = ""
        For i = 0 To UBound(m, 1)
            For j = 0 To UBound(m, 2)
                mensaje = mensaje & CStr(m(i, j)) & "   "
            Next
            mensaje = mensaje & vbNewLine 'añadir un salto de línea
        Next
        ImprimirMatriz = mensaje
    End Function
End Module
```

22 Longitud de palabras (grado de dificultad: 3)

22.1Enunciado

Realizar un programa en Visual Basic que solicite al usuario una serie de palabras y que permita averiguar que palabra es la mayor y que palabra contiene más vocales.

Desarrollo del programa:

* almacenaremos en un vector las palabras solicitadas, previa petición al usuario de cuantas va a introducir. Y almacenaremos en una matriz, en la primera fila la longitud de cada palabra introducida y la segunda fila la cantidad de vocales que tiene cada palabra.

El programa al ejecutarse debe de aparecer el siguiente menú:

“Elegir opción a realizar:

1.- Salir

2.- Solicitar palabras

3.- Indicar que palabra es la mayor y que longitud tiene

4.- Indicar que palabra tiene más vocales y cuantas tiene”

NOTA.- Si ha pulsado una opción distinta de las anteriores debe mostrar el siguiente mensaje: “Opción incorrecta” y volver a mostrar el menú.

Utilizar las siguientes funciones:

- La función Len([vble_tipo_string] Devuelve un entero que contiene el número de caracteres de una cadena
- La función Redim [vble_vector] (tamaño vector) crea un vector de elementos desde 0 hasta tamaño_vector
- La función Mid(vble_tipo_string, Posición_inicial, Cantidad_caracteres) devuelve de una vble tipo string, desde la posición inicial que se le indique, la cantidad de caracteres que se quieren extraer. Por ejemplo:

  Dim** palabra **As String = "Prueba"

  Msgbox(Mid(**palabra**, 1, 3)) ‘mostrará en pantalla “Pru”

  *Vector**(2)= Mid(**palabra**, 2, 1) ‘guarda en la posición 2 del vector la letra “r”***

22.2Solución

```
Module Longitud_palabra
    Sub main()
        Dim vector_palabras() As String
        Dim ctas_palabras As Byte
        Dim matriz_valores(,) As Byte
        Dim menu As Byte
        Dim max_vocales, max_longitud, contar_vocales As Byte
        Dim pos_max_vocales, pos_max_palabra As Byte
        Dim letra As Char = ""

        Do
            menu = InputBox("Elija una opcion:" & vbCrLf &
                            "1.- Salir" & vbCrLf &
```

```
                            "2.- Solicitar palabras" & vbCrLf &
                            "3.- Indicar que palabra es mayor y que longitud tiene" & vbCrLf
&
                            "4.- Indicar que palabra tiene más vocales y cuantas tiene")
            Select Case menu
                Case 1
                    MsgBox("Fin del programa")
                Case 2
                    Do
                        ctas_palabras = InputBox("Ctas palabras desea a introducir")
                        If ctas_palabras < 1 Then
                            MsgBox("Error: debe de indicar valores >=1")
                        End If
                    Loop While ctas_palabras < 1
                    'Redimensionamos el vector con la cantidad de palabras a introducir
                    ReDim vector_palabras(ctas_palabras - 1)
                    'Redimensionamos la matriz donde van a ir los cálculos
                    'La primera dimensión es 1, que serían dos filas
                    ReDim matriz_valores(1, ctas_palabras - 1)
                    'Pedimos las palabras y también ya guardamo su tamaño
                    For indice = 0 To UBound(vector_palabras)
                        vector_palabras(indice) = InputBox("Introduce la palabra nº " &
indice + 1)
                        'Guardamos en la primera fila de la matriz la longitud de cada
palabra
                        matriz_valores(0, indice) = Len(vector_palabras(indice))
                        'También podríamos ya calcular su vocales pero lo hacemos en el case
4
                    Next
                Case 3 'Palabra mayor y cual es
                    max_longitud = matriz_valores(0, 0)
                    pos_max_palabra = 0
                    'recorremos las columnas de la matriz para ver el valor mayor
                    'que esta almacenado en la primera fila, o sea, la cero
                    For columna = 0 To UBound(matriz_valores, 2)
                        If matriz_valores(0, columna) > max_longitud Then
                            max_longitud = matriz_valores(0, columna)
                            pos_max_palabra = columna
                        End If
                    Next
                    'La posicion de la palabra en el vector coincide con la posición en la
matriz de la columna
                    MsgBox("La palabra mayor es " & vector_palabras(pos_max_palabra) &
vbCrLf &
                           "Y su tamaño es " & matriz_valores(0, pos_max_palabra))
                Case 4 'Palabra con más vocales y ctas tiene
                    'Primero calculamos el total de vocales que tiene cada palabra
                    'con lo que recorremos cada palabra del vector para calcular sus vocales
                    For indice = 0 To UBound(vector_palabras)
                        'Creamos un ciclo para recorrer con la función MID cada letra de la
palabra
                        contar_vocales = 0 'contador de vocales
                        'Empezamos en "1" ya que son las posiciones de las letras
                        'de una palabra, no tiene que ver con el vector o la matriz de la
posicion "0"
                        For i = 1 To matriz_valores(0, indice) 'Hasta la longitud que ya
tenemos guardada
                            letra = Mid(vector_palabras(indice), i, 1)
                            Select Case letra
                                Case "a"
                                    contar_vocales = contar_vocales + 1
                                Case "e"
                                    contar_vocales = contar_vocales + 1
                                Case "i"
```

```
                                    contar_vocales = contar_vocales + 1
                                Case "o"
                                    contar_vocales = contar_vocales + 1
                                Case "u"
                                    contar_vocales = contar_vocales + 11
                            End Select
                        Next
                        'Guardar "0" si no encuentra vocales y si no la suma de vocales
encontradas
                        matriz_valores(1, indice) = contar_vocales
                    Next
                    'recorremos las columnas de la matriz para ver el valor mayor
                    'que esta almacenado en la segunda fila, o sea, la uno
                    max_vocales = matriz_valores(1, 0)
                    pos_max_vocales = 0
                    For columna = 0 To UBound(matriz_valores, 2)
                        If matriz_valores(1, columna) > max_vocales Then
                            max_vocales = matriz_valores(1, columna)
                            pos_max_vocales = columna
                        End If
                    Next
                    If max_vocales = 0 Then
                        MsgBox("No existen palabras con vocales guardadas")
                    Else
                        MsgBox("La palabra con más vocales es " &
vector_palabras(pos_max_vocales) & vbCrLf &
                                                          "Y tiene en total " & matriz_valores(1,
pos_max_vocales))
                    End If
                Case Else
                    MsgBox("Error: opción incorrecta")
            End Select
        Loop Until menu = 1
    End Sub
End Module
```

23 Programa para introducir las notas de los alumnos (evolución III) (grado de dificultad: 4)

23.1Enunciado

El profesor de la asignatura “Informática aplicada a la ingeniería” desea crear un programa que le permita introducir y almacenar las notas de los alumnos de todo el cuatrimestre. Se utilizarán ficheros.

Teniendo en cuenta:

1. Todos los alumnos son españoles (tienen NIF).
2. Habrá un menú, en donde el profesor decidirá: Introducir un alumno con sus notas, ver el listado completo con sus notas, consultar la nota de un alumno o salirse del programa.
3. El alumno que realiza evaluación continua, realizará 4 exámenes y su nota final es la media de las 4 pruebas realizadas. Si el alumno renuncia a la evaluación continua, su nota final será la que obtenga en el examen de mayo.
4. No se utilizarán vectores para vaciar los datos del archivo.
5. El directorio será una constante, por ejemplo c:\temp\alumnos.txt).

23.2Solución

```
Module Module1
  'Declaración de una nueva variable llamada alumno
  Structure alumno
    Public DNI As String
    Public nombre As String
    Public Evaluacion_continua As Boolean
    Public prueba1 As Double
    Public prueba2 As Double
    Public prueba3 As Double
    Public prueba4 As Double
    Public media As Double
  End Structure
  'Pulsar cualquier tecla para continuar
  Sub espera_intro(ByRef mensaje As String)
    Console.WriteLine(mensaje)
    Console.ReadLine()
  End Sub
  'Deja dos líneas en blanco
  Sub saltos()
    Console.WriteLine()
    Console.WriteLine()
  End Sub
  ' Borrar pantalla
  Sub limpiar()
    Console.Clear()
  End Sub
  ' Función permite comprobar que la nota introducuida sea número
  Function Comprobar_si_es_prueba(ByVal prueba As String) As Boolean
    If Val(prueba) >= 0 And Val(prueba) <= 10 Then
      Comprobar_si_es_prueba = True
    Else : Comprobar_si_es_prueba = False
      Console.WriteLine("La nota debe ser numérica y/o menor o igual que 10,
vuelve a introducirla")
    End If
  End Function
```

```
  'Función que permite verificar si el DNI contiene 8 números y una letra al
final
  Function comprobar_DNI(ByVal temp As String) As Boolean
    Dim Numero_caracteres As Integer
    Dim numero As String
    Dim letras As Char
    Numero_caracteres = Len(temp)
    numero = Mid(temp, 1, 8)
    letras = UCase(Mid(temp, 9, 9))
    If Numero_caracteres = 9 And IsNumeric(numero) And letras Like "[A-Z]" Then
      comprobar_DNI = True
    Else
      Console.WriteLine("DNI Incorrecto, debe contener 8 dígitos y una letra")
      comprobar_DNI = False
    End If
  End Function
  'Función que permite comrpobar que el nombre contenga solamente letras
  Function comprobar_nombre(ByVal temp As String) As Boolean
    Dim Numero_caracteres, i, numero_caracteres_c As Integer
    Dim letras As Char
    Numero_caracteres = Len(temp)
    letras = UCase(Mid(temp, 1, Numero_caracteres))
    i = 1
    numero_caracteres_c = 0
    Do
      letras = UCase(Mid(temp, i, i))
      If letras Like "[A-Z]" Or letras = Chr(32) Then
        numero_caracteres_c = numero_caracteres_c + 1
      End If
      i = i + 1
    Loop Until i = Numero_caracteres + 1
    If numero_caracteres_c = Numero_caracteres Then
      comprobar_nombre = True
    Else
      comprobar_nombre = False
    End If
  End Function
  'Pedir los datos de un alumno y verificarlos
  Sub pedir_datos(ByRef matriz As alumno)
    Dim continua As Char
    Dim Temp As String
    Dim cierto As Boolean
    With matriz
      cierto = False
      Do
        Console.Write("Dame el DNI del alumno ===>")
        Temp = Console.ReadLine()
        Temp = UCase(Trim(Temp))
      Loop Until comprobar_DNI(Temp) = True
      .DNI = Temp
      Do
        Console.Write("Dame el Nombre del alumno ===>")
        Temp = Console.ReadLine()
      Loop Until comprobar_nombre(Temp) = True
      .nombre = Temp
      .Evaluacion_continua = True
      Console.Write("Renuncia a la evaluación continua (Teclee S, cualquier otro
caracter es evaluación continua): ")
      continua = Console.ReadLine()
      If UCase(continua) = "S" Then
        .Evaluacion_continua = False
      End If
      If .Evaluacion_continua = True Then
```

```
        Do
          Console.Write("Dame la nota del primer examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba1 = Val(Temp)
        Do
          Console.Write("Dame la nota del segundo examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba2 = Val(Temp)
        Do
          Console.Write("Dame la nota del tercer examen: ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba3 = Val(Temp)
        Do
          Console.Write("Dame la nota del último examen (Fecha Oficial Mayo): ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba4 = Val(Temp)
      End If
      If .Evaluacion_continua = False Then
        Do
          Console.Write("Dame la Nota del examen (Fecha Oficial Mayo): ")
          Temp = Console.ReadLine()
        Loop Until Comprobar_si_es_prueba(Temp) = True
        .prueba1 = 0
        .prueba2 = 0
        .prueba3 = 0
        .prueba4 = 0
        .media = Val(Temp)
      Else
        If (.prueba1 + .prueba2 + .prueba3 + .prueba4) = 0 Then
          .media = 0
        Else
          .media = Math.Round((.prueba1 + .prueba2 + .prueba3 + .prueba4) / 4,
2)
        End If
      End If
    End With
  End Sub
  Function Pedir_DNI() As String
    Dim temp As String
    Console.Write("Dame el NIF del alumno ===>")
    temp = Console.ReadLine()
    temp = UCase(Trim(temp))
    Pedir_DNI = temp
  End Function
  ' Permite imprimir todas las calificaciones de los alumnos (genera u listado).
  Sub Imprimir_datos(ByVal matriz As alumno)
    Dim imprimir As String
    With matriz
      If .Evaluacion_continua = True Then
        imprimir = .DNI & vbTab & .nombre & vbTab & vbTab & .prueba1 & vbTab &
.prueba2 & vbTab & .prueba3 & vbTab & .prueba4 & vbTab & .media
      Else
        imprimir = .DNI & vbTab & .nombre & vbTab & vbTab & " " & vbTab & " " &
vbTab & " " & vbTab & " " & vbTab & .media
      End If
      Console.Write(imprimir)
    End With
    Console.WriteLine()
  End Sub
```

```
  Sub menu()
    saltos()
    Console.WriteLine("Menú")
    Console.WriteLine("1. Introducir alumno")
    Console.WriteLine("2. Listar todos los alumnos")
    Console.WriteLine("3. Consultar las notas de un alumno")
    Console.WriteLine("4. Salir")
    Console.Write("Cuál es tu opción ====> ")
  End Sub
  Sub Main()
    'Declaración de variables
    Dim DNI_buscar As String
    Dim alumno_encontrado As alumno
    Dim matriz As alumno
    Dim encontrar As Boolean
    Dim opcion As Byte
    Dim File As String
    'Variables que se utilizan en archivos
    Dim idfichero As Integer
    Dim nombre As String

    nombre = "c:\temp\alumno.txt"
    File = (nombre)
    Console.WriteLine("File" & File)
    'Crear un menú que permite modificar una nota, imprimir los alumnos o salir
del program
    Do
      menu()
      opcion = Console.ReadLine()
      Select Case opcion
        Case 1
          limpiar()
          'introducir un alumno
          idfichero = FreeFile()
          If StrComp(File, "") = 0 Then
            Console.WriteLine("El archivo no ha sido creado")
            FileOpen(idfichero, nombre, OpenMode.Output)
          Else
            FileOpen(idfichero, nombre, OpenMode.Append)
          End If
          pedir_datos(matriz)
          With matriz
            Write(idfichero, .DNI, .nombre, .Evaluacion_continua, .prueba1,
.prueba2, .prueba3, .prueba4, .media)
          End With
          FileClose(idfichero)
        Case 2
          'Listado de los alumnos que existen con sus calificaciones
          limpiar()
          If StrComp(File, "") = 0 Then
            Console.WriteLine("No exite listado de alumnos")
          Else
            idfichero = FreeFile()
            FileOpen(idfichero, nombre, OpenMode.Input)
            Console.WriteLine("LISTADO DE ALUMNOS")
            Console.WriteLine("NIF " & vbTab & vbTab & "Nombre " & vbTab & vbTab
& "Nota 1" & vbTab & "Nota 2" & vbTab & "Nota 3" & vbTab & "Nota 4" & vbTab &
"Media")
            While Not EOF(idfichero)
              With matriz
                Input(idfichero, .DNI)
                Input(idfichero, .nombre)
                Input(idfichero, .Evaluacion_continua)
```

```
                Input(idfichero, .prueba1)
                Input(idfichero, .prueba2)
                Input(idfichero, .prueba3)
                Input(idfichero, .prueba4)
                Input(idfichero, .media)
              End With
              Imprimir_datos(matriz)
            End While
            FileClose(idfichero)
            espera_intro("teclea <Intro> para continuar")
            limpiar()
          End If
        Case 3
          limpiar()
          Console.WriteLine("Consultar la calificacion de un alumno")
          encontrar = False
          If StrComp(File, "") = 0 Then
            Console.WriteLine("No exite listado de alumnos")
          Else
            idfichero = FreeFile()
            FileOpen(idfichero, nombre, OpenMode.Input)
            DNI_buscar = Pedir_DNI()
            While Not EOF(idfichero)
              With matriz
                Input(idfichero, .DNI)
                Input(idfichero, .nombre)
                Input(idfichero, .Evaluacion_continua)
                Input(idfichero, .prueba1)
                Input(idfichero, .prueba2)
                Input(idfichero, .prueba3)
                Input(idfichero, .prueba4)
                Input(idfichero, .media)
              End With
              If DNI_buscar = matriz.DNI Then
                alumno_encontrado = matriz
                Console.WriteLine("encontro")
                encontrar = True
              End If
            End While
            If encontrar = True Then
              Console.WriteLine("NIF " & vbTab & vbTab & "Nombre " & vbTab &
vbTab & "Nota 1" & vbTab & "Nota 2" & vbTab & "Nota 3" & vbTab & "Nota 4" &
vbTab & "Media")
              Imprimir_datos(alumno_encontrado)
            Else
              encontrar = False
              Console.WriteLine("No existe en el archivo un alumno con ese NIF")
            End If
            espera_intro("teclea <Intro> para salir")
            FileClose(idfichero)
          End If
      End Select
    Loop Until opcion = 4
    limpiar()
    espera_intro("teclea <Intro> para salir")
  End Sub
End Module
```

24 Ejercicio de escritura / lectura de datos vector en un fichero (grado de dificultad: 4)

24.1 Enunciado

Realizar un programa que le permita a un usuario genera números aleatorios en el rango [-20, 200] hasta un máximo de 100 números, guardarlos en un fichero y poder leerlos para mostrarlos y calcular el valor máximo leido.

El programa mostrará un menú inicial con las siguientes opciones:

0.- Fin del programa

1.- Generar datos aleatorios y almacenar en un fichero

2.- Leer datos del fichero y trabajar con ellos

- *Opción 0*, finalizará el programa y mostrará el siguiente mensaje:

  "FIN DEL PROGRAMA: Fichero con datos simples"

- *Opción 1*, mostrará el siguiente submenú:

  0.- Volver al menú principal

  1.- Añadir datos al fichero existente

  2.- Sobrescribir los datos existentes

  - Opción 0, vuelve al menú principal

  NOTA.- LAS DOS OPCIONES -1 y 2- VAN EN EL MISMO CASE, YA QUE LA DIFERENCIA SÓLO ESTÁ EN EL MODO DE ABRIR EL FICHERO PARA ESCRITURA

  - Opción 1 to 2:
    - Opción 1.- abre el fichero que se define en el programa en la vble "ruta_vector" y añade al final del mismo los datos.
    - Opción 2.- abre el fichero que se define en el programa en la vble "ruta_vectror" y sobrescribe los datos que existen (en concreto borra lo que tiene y escribe lo nuevo).

  A continuación se le pide al usuario cuantos números quiere generar hasta un máximo de 100. A continuación se redimensiona el vector, y se llama a un procedimiento "rellenar_aleatorios_vector" que se le pasan tres parámetros: el vector donde guardara los valores aleatorios, el valor mínimo y el valor máximo del rango de los números aleatorios a calcular. Una vez tiene los valores ya guardados en el vector (que al ser pasado por Byref, automáticamente ya los está guardando), se escriben en el fichero mediante un ciclo que va recorriendo los valores del vector.

- *Opción 2*, comprobará primero si existe el fichero del cual leer, que es el que está indicado en la vble "ruta_vector" y de no existir muestra el siguiente mensaje de error: "Error: No existe fichero de datos."

  En el caso de existir dicho fichero, abrimos el fichero para lectura que se define en la vlbe "ruta_vector" y leemos los datos del fichero en una nueva vble "vector_leido". Comprobamos que el fichero tiene datos, comprobando si la vble "índice" es igual a 0 el fichero no tenía datos y mostramos el mensaje: "El fichero no tiene datos para leer", en caso contraio mostrará el siguiente submenú:

  0.- Volver al menú principal

1.- Mostrar los valores almacenados en el vector leído

2.- Calcular el valor máximo del vector leído

- Opción 0, vuelve al menú principal
- Opción 1, Mostramos los datos del vector almacenados en un string. Para ello llamamos a la función "mostar_datos_vector" que se le pasa el vector y nos devuelve en un string los datos del vector
- Opción 2, Mostramos el valor máximo del vector leído. Para ello llamamos a la función "mostrar_maximo_fichero" que se le pasa el vector leído y devuelve el valor máximo de dicho vector.

24.2Solución

```
Module fichero_datos_vector
    Sub main()
        Dim vector_aleatorio() As Integer 'Vector que guardará los valores aleatorios
generados
        Dim cantidad As Integer 'Almacena cuantos números aleatorios quiere que se generen,
introducido por el usuario
        Const rango_minimo As Integer = -20 'Valor mínimo a generar aletatoriamente
        Const rango_maximo As Integer = 200 'Valor máximo a generar aleatoriamente
        Const max_cantidad As SByte = 100 'Cantidad máxima permitida de números aleatorios a
generar
        Dim vector_leido() As Integer 'Vector que almacenará los valores leídos del fichero
        Dim cad_vector As String = "" 'Cadena que guarda los datos del vector_leido para
mostarlos al usuario

        Dim idfichero_vector As Integer 'Vble que guarda el integer que identificará el
archivo a utilizar
        'Ruta del archivo con su nombre (por defecto en la carpeta del proyecto /Bin/Debug)
        Dim ruta_vector As String = "Vector.txt"


        Dim opciones_validas As String = "" 'para los case else de los menus, ver Sub
Msg_error (byval op_ok as string)
        Dim menu As Byte 'Guarda la selección de los menus
        Dim submenu As Byte 'Guarda la selección de los submenus
        'Definimos "indice" de tipo integer porque en uno de los casos
        'al leer los datos de un fichero puede tener más de 256 elementos (tamaño máximo de
un byte)
        Dim indice As Integer = 0 'Vble que recorre las posiciones de los elementos de un
vector
        Randomize()

        idfichero_vector = FreeFile()

        Do
            menu = InputBox("Elija una opción:" & vbCrLf &
                            "0.- Fin del programa" & vbCrLf &
                            "1.- Generar datos aleatorios y almacenar en un fichero" & vbCrLf
&
                            "2.- Leer datos del fichero y trabajar con ellos", "Ficheros con
datos de un vector")
            Select Case menu
                Case 0
                    MsgBox("FIN DEL PROGRAMA: Fichero con datos de un vector",
MsgBoxStyle.Information, "Ficheros con datos de un vector")
                Case 1 'Pedir datos y almacenarlos en un fichero
                    Do
                        submenu = InputBox("Elija una opción:" & vbCrLf &
                                           "0.- Volver al menú principal" & vbCrLf &
```

```
                                            "1.- Añadir datos al fichero existente" &
vbCrLf &
                                            "2.- Sobrescribir los datos existentes",
"Escribir en el fichero")
                        'Comprobamos si existe el fichero para lectura
                        Select Case submenu
                            Case 0 'Volver al menú anterior
                                'MsgBox("Opción cancelada", MsgBoxStyle.Information,
"Escribir datos en el fichero")
                            Case 1 To 2 'Abrimos el fichero para escribir
                                'Comprobamos que opción ha escogido el usuario para escribir
en el fichero los datos
                                If submenu = 1 Then
                                    'Abrimos el fichero para añadir datos al final
                                    FileOpen(idfichero_vector, ruta_vector, OpenMode.Append)
                                Else
                                    'Abrimos el fichero y escribe en la primera posición,
sobresescribiendo los datos que existan
                                    FileOpen(idfichero_vector, ruta_vector, OpenMode.Output)
                                End If
                                'Preguntamos al usuario cuantos números quiere generar
aleatoriamente
                                Do
                                    cantidad = InputBox("Cuantos valores aleatorios
generamos? (Máximo " & max_cantidad & " números)", "Generar números aleatorios")
                                    If cantidad <= 0 Or cantidad > max_cantidad Then
                                        MsgBox("ERROR: Máximo de números a generar " &
max_cantidad & vbCrLf &
                                               "Y minimo 1", MsgBoxStyle.Exclamation,
"Error dato introducido")
                                    End If
                                Loop Until cantidad > 0 And cantidad <= max_cantidad
                                'Se crea el vector con el tamaño indicado
                                ReDim vector_aleatorio(cantidad - 1)
                                'Se rellena el vector con los datos aleatorios
                                rellenar_aleatorios_vector(vector_aleatorio, rango_minimo,
rango_maximo)
                                'Guardamos en el fichero los datos del vector calculado
                                'para ello vamos recorriendo del vector y guardando cada
valor en el fichero
                                For indice = 0 To UBound(vector_aleatorio)
                                    'Escribimos en el fichero cada valor del elemento del
vector
                                    Write(idfichero_vector, vector_aleatorio(indice))
                                Next
                                FileClose() 'Cerramos el fichero
                            Case Else 'Cualquier opción no válida del submenu
                                opciones_validas = "0, 1 y 2"
                                msg_error(opciones_validas)
                        End Select
                    Loop Until submenu = 0
                Case 2 'Abrir el fichero y leer los datos
                    'Comprobamos si existe el fichero para lectura
                    'My.computer.filesystem.fileExist devuelve "TRUE" si existe el fichero
con el nombre que se le pasa
                    If My.Computer.FileSystem.FileExists(ruta_vector) = False Then
                        MsgBox("Error: No existe fichero de datos.",
MsgBoxStyle.Exclamation, "Leer fichero")
                    Else 'El fichero existe y lo abrimos para lectura
                        'Abrimos el fichero para lectura
                        FileOpen(idfichero_vector, ruta_vector, OpenMode.Input)
                        'Almacenamos los datos en la vble "vector_leido"
                        'Inicializamos el indice del vector
                        indice = 0
```

```
                        'Recorremos el fichero desde el principio hasta el final del fichero
(EOF)
                        Do While Not EOF(idfichero_vector)
                            'Creamos el vector con el tamaño de los datos que se van leyendo
                            '-Preserve- manteniendo los que ya existen para no
sobreescribirlos cada vez que se crea
                            ReDim Preserve vector_leido(indice)
                            'Guardamos en cada posición del vector los números leídos
                            Input(idfichero_vector, vector_leido(indice))
                            indice += 1 'vamos incrementando el indice de los datos a leer
                        Loop
                        FileClose() 'Cerramos el fichero

                        'Comprobamos si se ha leido algún dato
                        If indice = 0 Then 'Fichero sin datos para leer
                            MsgBox("No hay datos en el fichero para leer", "Leer fichero")
                        Else 'El fichero tiene datos
                            'Mostrarmos el submenu
                            Do
                                submenu = InputBox("Elija una opción:" & vbCrLf &
                                                   "0.- Volver al menú principal" & vbCrLf
&
                                                   "1.- Mostrar valores almacenados en el
vector leido" & vbCrLf &
                                                   "2.- Calcular el valor máximo del vector
leido", "Leer datos en fichero")
                                Select Case submenu
                                    Case 0 'Volver al menú anterior
                                        'MsgBox("Opción cancelada", MsgBoxStyle.Information,
"Leer datos del fichero")
                                    Case 1 'Mostrar valores almacenados en el vector
                                        MsgBox("Los datos leidos son:" & vbCrLf &
mostrar_datos_vector(vector_leido))
                                    Case 2 'valor máximo del vector
                                        MsgBox("El valor máximo leido del fichero es: " &
mostrar_maximo_fichero(vector_leido))
                                    Case Else 'Cualquier opción no válida del submenu
                                        opciones_validas = "0, 1 y 2"
                                        msg_error(opciones_validas)
                                End Select
                            Loop Until submenu = 0
                        End If
                    End If
                Case Else 'Cualquier opción no válida del menu
                    opciones_validas = "0, 1 o 2"
                    msg_error(opciones_validas)
            End Select
        Loop Until menu = 0
    End Sub
    Sub msg_error(ByVal op_ok As String) 'No devuelve nada, sólo muestra un mensaje
        'Muestar un mensaje de error para el case else
        'pasandole las opciones correctas a seleccionar por el usuario

        MsgBox("ERROR: Opcion incorrecta. Opciones válidas: " & op_ok, MsgBoxStyle.Critical,
"ERROR opcion")
    End Sub
    Sub rellenar_aleatorios_vector(ByRef vector_original() As Integer, ByVal rango_minimo As
Integer, ByVal rango_maximo As Integer)
        'Los vectores son pasados siempre por referencia, aunque se ponga Byval
        'se guardan directamente los valores en el vector pasado y por lo tanto no hay que
devolver nada
        'NOTA.- los valores del vector ya están automáticamente en la vble "vector" del sub
main
        Dim indice As Byte
```

```
        For indice = 0 To UBound(vector_original)
            vector_original(indice) = rango_minimo + Int(Rnd() * (rango_maximo -
rango_minimo + 1))
        Next
    End Sub
    Function mostrar_datos_vector(ByRef v_leido_copia() As Integer) As String
        'Devuelve en una cadena los valores del vector pasado cómo parámetro
        'Definimos la cadena que almacenará los valores del vector
        Dim cad_vector As String = ""
        'Recorremos los elementos del vector para guardar sus valores
        For i = 0 To UBound(v_leido_copia)
            cad_vector = cad_vector & v_leido_copia(i) & " "
        Next
        'Devolvemos la cadena con los datos del vector
        Return cad_vector
    End Function
    Function mostrar_maximo_fichero(ByRef v_leido_copia() As Integer) As Integer
        'Devuelve el valor máximo de los datos del vector de enteros
        'Declaramos la vble que guardará el máximo de los datos del vector
        Dim maximo As Integer
        'Iniciamos el máximo al rprimer valor del vector
        maximo = v_leido_copia(0)
        'Recorremos todos los valores del vector y los comparamos con la vble máximo
        For i = 0 To UBound(v_leido_copia)
            If v_leido_copia(i) > maximo Then 'si el valor del vector es mayor que el del
máximo
                maximo = v_leido_copia(i) 'guardamos en el máximo el valor del dato del
vector
            End If
        Next
        'Devolvemos el valor máximo
        Return maximo
    End Function
End Module
```

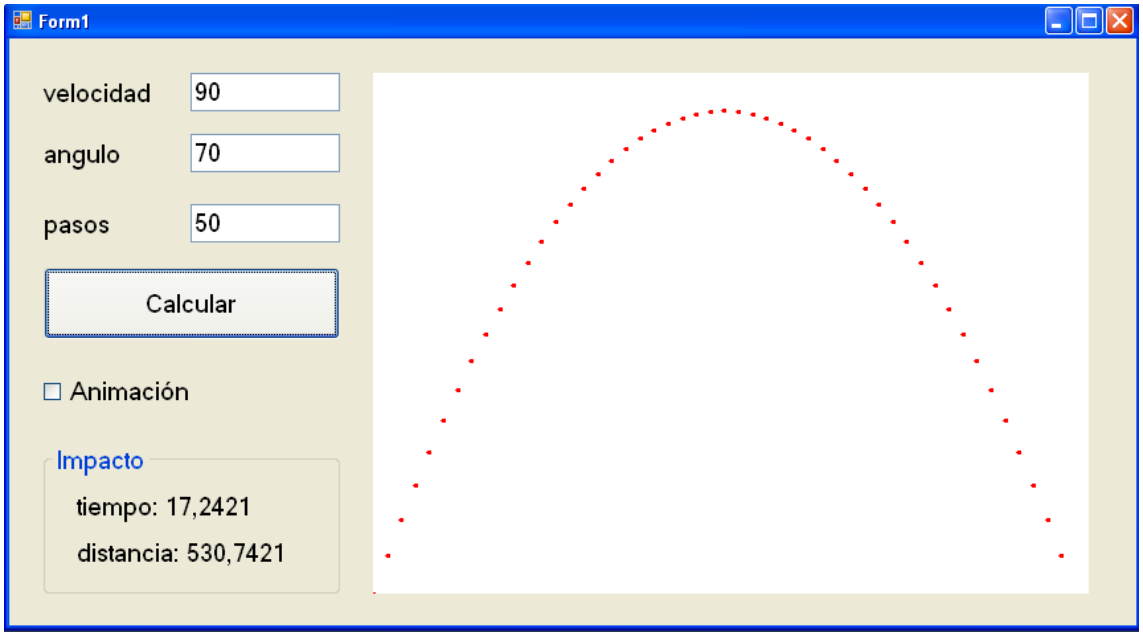
25 Balística Gráfico (grado de dificultad: 4)

25.1Enunciado

Desarrollar un programa capaz de calcular y visualizar en un gráfico la trayectoria del proyectil dadas una velocidad, ángulo de disparo y tiempo de simulación.

El programa deberá:

- Definir constantes y variables.
- Implementar un interfaz gráfico basado en formularios como el que se muestra a continuación para solicitar los parámetros al usuario y mostrar los resultados.
- Cuando el usuario pulse un botón "Calcular", calcular el tiempo y distancia hasta el impacto (y=0) y actualizar el gráfico. El parámetro "pasos" corresponde al número de instantes que se tomarán desde el instante cero hasta el instante del impacto.
- En el caso de que el CheckBox "Animación" esté pulsado, se deberá utilizar un Timer para mostrar la trayectoria punto por punto a lo largo del tiempo.



25.2Solución

```
Public Class Form1
    Dim Vo, ang, dt As Double
    Dim timpacto, ximpacto As Double
    Const g As Double = 9.81
    Dim pasos, pact As Integer
    Dim x(), y() As Double
    Dim t() As Double
    Private Sub ButtonCalcular_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonCalcular.Click
        Try
            Vo = TextBoxVo.Text
            ang = TextBoxAng.Text * 2 * Math.PI / 360.0
            pasos = TextBoxPasos.Text
            timpacto = Vo * Math.Sin(ang) / (0.5 * g) ' no voy a simular más tiempo
```

```
            ximpacto = Vo * Math.Cos(ang) * timpacto
            dt = timpacto / pasos

            LabelTimpacto.Text = "tiempo: " & FormatNumber(timpacto, 4)
            LabelDistancia.Text = "distancia: " & FormatNumber(ximpacto, 4)
            calculaTrayectoria()
            pact = 0
            If CheckBoxAnima.Checked Then
                TimerAnima.Interval = 1000 * dt 'tiene que estar en milisegundos
                TimerAnima.Start()
            Else
                Panel1.Refresh()
            End If
        Catch ex As Exception
            MsgBox("Error, deben ser números")
        End Try
    End Sub

    Private Sub calculaTrayectoria()
        Dim Xo = 0.0, Yo = 0.0
        ReDim x(pasos - 1)
        ReDim y(pasos - 1)
        ReDim t(pasos - 1)

        If Vo <> 0.0 Then
            For i = 0 To pasos - 1
                t(i) = i * dt
                x(i) = Xo + Vo * Math.Cos(ang) * t(i)
                y(i) = Yo + Vo * Math.Sin(ang) * t(i) - 0.5 * g * (t(i) ^ 2)
            Next
        End If
    End Sub

    Private Sub TimerAnima_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimerAnima.Tick
        If pact < pasos - 1 Then
            Panel1.Refresh()
            pact += 1
        Else
            Panel1.Refresh()
            pact = 0
            TimerAnima.Stop()
        End If
    End Sub

    Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint
        Dim Ymax As Integer = Panel1.Size.Height
        Dim xi, yi As Integer
        If pasos > 0 And pact < pasos Then
            If CheckBoxAnima.Checked Then
                xi = x(pact)
                yi = y(pact)
                e.Graphics.FillEllipse(Brushes.Red, xi - 2, Ymax - yi - 2, 4, 4)
            Else
                For i = 0 To pasos - 1
                    xi = x(i)
                    yi = y(i)
                    e.Graphics.FillEllipse(Brushes.Red, xi - 2, Ymax - yi - 2, 4, 4)
                Next
            End If
        End If
    End Sub
End Class
```