COLEGIO VILLA RICA IED

Guía No 3

Área de Informática **Periodo 3 Octubre** **Grado Octavo**

Docente Julian David Calderón Burgos

Nombre del Estudiante:_____Curso:_____ Fecha:_____

Unidad Temática: Introducción a los Programas de Presentación

Propósito: Manejo de conceptos básicos del programa de presentaciones e indagación de nuevos dispositivos

Descripción de la Guía: La siguiente guía presenta la introducción sobre los programas de presentación y su importancia.

Presentación del tema:

Realice un dibujo que este en una hoja completa y que diga Tercer Periodo Académico con algún elemento relacionado con la tecnología, luego copie la siguiente información de Inicio del Periodo.

Eje Temático:

Lenguajes de programación c++ y visual Basic

Estándar:

En el mundo actual, se señala la alfabetización científica y tecnológica como una necesidad inaplazable, en tanto se espera que todos los individuos estén en capacidad para acceder, utilizar, evaluar, y transformar artefactos, procesos y sistemas tecnológicos para la vida social y productiva.

Aprendizajes Básicos DBA

Elaboración de programas básicos utilizando el lenguaje c++

Comprensión de la sintaxis de un programa de Visual Basic o Lenguaje c.

VARIABLES EN VISUAL BASIC

La forma de declarar las variables es la siguiente:

Dim| Public| Static nombre_variable As tipo

Dim: Al declarar una variable con esta palabra estamos diciendo que la variable sea local al ámbito en que se declara. Puede ser dentro de un procedimiento o dentro de un formulario, de esta forma no sería accesible desde los demás procedimientos o formularios.

Public: Las variables declaradas serán públicas y podrán estar accesibles desde todos los formularios de la aplicación. Para conseguirlo tendremos que declararlas en un módulo de código, no en la sección declarations de cualquier formulario de los que conste la aplicación. Para crear un módulo de código en el menú principal de Visual Basic marcamos en INSERT/MODULE y aparecerá junto a los demás formularios de la ventana de proyecto, aunque con un icono distinto indicando que se trata de un módulo de código.

Static: Con esta forma de declarar variables conseguiremos que las variables locales no se creen y se destruyan al entrar y salir de los procedimientos donde fueron declaradas, sino que se mantenga su valor durante todo el periodo de ejecución de la aplicación. De esta forma a entrar en algún procedimiento las variables recuerdan el valor que tenían cuando se salió de él.

TIPOS DE VARIABLES

TIPO	COMENTARIO
BOOLEAN	Sólo admite 2 valores TRUE o FALSE
BYTE	admite valores entre 0 y 255
INTEGER	admite valores entre -32768 y 32767
LONG	admite valores entre -2.147.483.648 y 2.147.483.647
SINGLE	admite valores decimales con precisión simple
DOUBLE	admite valores decimales de doble precisión
CURRENCY	válido para valores de tipo moneda
STRING	cadenas de caracteres

DATE	fechas, permite operar con ellas

ESTRUCTURAS DE SELECCIÓN

Disponemos de 2 tipos de sentencias para realizar selecciones dependiendo del valor de una expresión o una sola variable:

Sentencia If Then Else

La primera es la sentencia If Then Else que no debe tener secretos puesto que es muy similar en todos los lenguajes

La estructura general es la siguiente:

If condición then

bloque de sentencias

Else

bloque de sentencias

End If

Se pueden colocar todas las sentencias de código que queramos en cada uno de los bloques de sentencias, siempre que cada sentencia vaya en una línea distinta.

Si los bloques de sentencias están formados por una sola instrucción podemos utilizar la versión reducida que ocupa una sola línea:

If condición then sentencia else sentencia

Ejemplo:

If isnumeric(numero) then la variable número es numérica else no es numérica

También podemos anidar varias sentencias If then Else cuando una de las 2 opciones iniciales contiene a su vez 2 bifurcaciones dependiendo del estado de otra condición:

if dato<10 then

la variable dato contiene un solo dígito

Else

If dato<100 then

la variable dato contiene 2 dígitos

Else

la variable dato contiene más de 2 dígitos

end If

End If

En el ejemplo anterior se quiere saber si la variable dato contiene uno, dos o más dígitos. Para calcularlo no es suficiente con una sola sentencia If Then Else, por tanto, se recurre a anidar 2 sentencias de este tipo. La segunda sentencia If se ejecutará si la condición de la primera sentencia If no se cumple, es decir, si dato>=10. Una vez que se ha llegado a este punto sabemos que la variable dato contiene 2 o más dígitos, mediante una nueva sentencia If sabremos exactamente el número de dígitos de la variable dato.

Este último ejemplo podría haberse escrito de otra forma utilizando la cláusula ElseIf:

If dato<10 then

la variable dato contiene un dígito

ElseIf dato<100 then

la variable dato contiene dos digitos

Else

la variable dato contiene más de 2 digitos

End If

Esta segunda opción es perfectamente válida cuando queremos evaluar varias condiciones, aunque tiene más limitaciones que la estructura anterior ya que enlaza directamente el Else con el If siguiente, sin dejarnos introducir sentencias entre medias que algunas veces podríamos necesitar.

Podemos introducir tantas líneas ElseIF como queramos siempre antes del último Else, si es que lo necesitamos.

Para múltiples decisiones en los que dependiendo del valor de una variable queremos que se realice una acción distinta, parecido a los menús de los programas de MS DOS, no conviene utilizar la estructura IF Then, sino que el código queda más claro y resulta más fácil de modificar utilizando la sentencia Select

Estructura SELECT CASE

Esta sentencia permite realizar operaciones diferentes dependiendo del valor de una variable:

Estructura General:

Select Case dato

Case valor1

bloque de sentencias

case valor2

bloque de sentencias

Case valor3

bloque de sentencias

case else

bloque de sentencias, se ejecutan si no se cumple ninguno de los valores anteriores

End Select

En esta construcción, dependiendo del valor de la variable dato se ejecutará un bloque de sentencias diferente. Los valores que podemos colocar en lugar de valor1, valor2, valor3 no sólo se limitan a valores constantes como números y cadenas de texto, sino que podemos comparar con un número como podemos ver en el siguiente ejemplo:

Select Case NotaFinal

Case Is < 5

Suspendido

Case 5 to 6.99

Aprobado

Case 7 to 8.99

Notable

Case Else

Sobresaliente

End Select

Como podéis observar si utilizamos los operadores lógicos como >, <, =, <=, >= debemos anteponer el operador Is. Si lo que hacemos es comparar con un intervalo de valores colocaremos el operador to entre los límites del intervalo. También podemos realizar comparaciones con un conjunto de valores separados por comas:

Case 1, 3, 5.

el número es impar

Para terminar con el tema de las sentencias de selección vamos a ver un ejemplo completo en el que probaremos el uso de este tipo de instrucciones. Se trata del típico ejemplo de resolución de una ecuación de 2º grado

Sabemos que la estructura de una ecuación de este tipo es la siguiente:

$$ax^2 + bx + c = 0$$

La fórmula que resuelve el valor de x es:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Esta fórmula tiene 2 soluciones, una o ninguna dependiendo del contenido de la raíz, de modo que esa es la comparación que realizaremos:

Trabajaremos con variables de tipo double que permiten decimales con la mayor precisión:

```
Dim raiz, x1, x2, a, b, c As Double

raiz = b * b - 4 * a * c
Select Case raiz
Case Is < 0
    ' no tiene solución
Case 0
    'solo hay una solución
    x1 = (-b) / (2 * a)
Case Else
    'tiene 2 soluciones
    x1 = (-b + Sqr(raiz)) / (2 * a)
    x2 = (-b - Sqr(raiz)) / (2 * a)
End Select
```

También podría haberse utilizado la estructura If then Else, aunque de esta forma no hay que repetir tantas veces la condición a evaluar.

El código anterior es sólo una muestra de cómo llegar a la solución de un problema utilizando sentencias de Visual Basic, no hemos entrado todavía en como introducir este código en el entorno de desarrollo de VB para crear una aplicación. Eso lo veremos en el capítulo siguiente, donde hablaremos de ***controles*** que son los

elementos necesarios para construir el interfaz de usuario y poder asociar el código necesario para realizar nuestras aplicaciones.

CUESTIONARIO

1. ¿En visual basic para que sirve la palabra Dim?

2. ¿Cuándo se declara una variable con la palabra public quiere decir?

3. ¿Para qué sirve la instrucción static?

4. Realice un cuadro donde se nombren cada una de las variables que se pueden utilizar en visual basic y para que sirve cada una de ellas.

5. ¿Describa cómo se utiliza y para qué sirve la función if then else?

6. ¿Cuál es la diferencia entre if then else y la instrucción case?

7. ¿Cómo funciona la instrucción case?

Aprende en Casa
Desarrollado Área Informática
Docente Julian David Calderón Burgos