



aprenderaprogramar.com

# Ejercicios con Rnd y Randomize. Ordenación del código. Dividir líneas en Visual Basic. Ejemplos. (CU00316A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº15 del Curso Visual Basic Nivel I

29

## EJERCICIO CON RND Y RANDOMIZE

Vamos a generar un programa en Visual Basic donde utilizaremos el comando Rnd para generación de números aleatorios. El ejercicio es el siguiente: crear el código de un programa en el que se declaran tres variables tipo Single A, B y C cuyo valor se muestra en pantalla y debe estar comprendido entre cero y 10, excluido el diez.

*Solución (versiones VB menos recientes):*

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A!, B!, C As Single

Private Sub Form_Load()
Randomize
A = Rnd * 10
B = Rnd * 10
C = Rnd * 10
MsgBox ("A vale " & A & " ;B vale " & B & " ; C vale
" & C)
End Sub
```

*Solución (versiones VB más recientes):*

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A, B, C As Single

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Randomize()
        A = Rnd() * 10
        B = Rnd() * 10
        C = Rnd() * 10
        MsgBox("A vale " & A & " ;B vale " & B &
" ; C vale " & C)
    End Sub
End Class
```

## EJERCICIO NÚMEROS ALEATORIOS

Crear el código de un programa que genera tres números enteros aleatorios A, B, C comprendidos entre 65 y 90, ambos inclusive, y los muestra en pantalla.

*Solución (versiones VB menos recientes):*

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%, B%, C As Integer

Private Sub Form_Load()
Randomize
A = Int(Rnd * 26) + 65
B = Int(Rnd * 26) + 65
C = Int(Rnd * 26) + 65
MsgBox ("A vale " & A & " ;B vale " & B & " ; C vale
" & C)
End Sub
```

*Solución (versiones VB más recientes):*

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A, B, C As Integer

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Randomize()
        A = Int(Rnd() * 26) + 65
        B = Int(Rnd() * 26) + 65
        C = Int(Rnd() * 26) + 65
        MsgBox("A vale " & A & " ;B vale " & B &
" ; C vale " & C)
    End Sub
End Class
```

**Comentarios:** Si quieres respetar la literalidad de la fórmula que expusimos en apartados anteriores del curso puedes escribir con igual resultado lo siguiente:

$A = \text{Int}((90 - 65 + 1) * \text{Rnd} + 65)$ ;  $B = \text{Int}((90 - 65 + 1) * \text{Rnd} + 65)$ ;  $C = \text{Int}((90 - 65 + 1) * \text{Rnd} + 65)$

## CONCATENACIÓN DE ÓRDENES Y ORDENACIÓN SIMBÓLICA DEL CÓDIGO

La escritura de órdenes con Visual Basic se realiza con los criterios que ya hemos visto: las instrucciones pueden escribirse una debajo de otra en líneas independientes o bien en una misma línea separadas por dos puntos. Así estos dos códigos resultan equivalentes:

**Código V1 (versiones VB menos recientes):**

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%, B%, C%, D%

Private Sub Form_Load()
A = 3: B = 2: C = A * B: D = C ^ 2
MsgBox ("D vale " & D)
End Sub
```

**Código V1 (versiones VB más recientes):**

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A, B, C, D%

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        A = 3 : B = 2 : C = A * B : D = C ^ 2
        MsgBox("D vale " & D)
    End Sub
End Class
```

**Código V2 (versiones VB menos recientes):**

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%, B%, C%, D%

Private Sub Form_Load()
A = 3
B = 2
C = A * B
D = C ^ 2
MsgBox ("D vale " & D)
End Sub
```

**Código V2 (versiones VB más recientes):**

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A, B, C, D%

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        A = 3
        B = 2
        C = A * B
        D = C ^ 2
        MsgBox("D vale " & D)
    End Sub
End Class
```

La sangría al escribir código la estableceremos normalmente a través del tabulador, aunque para algunas instrucciones puede ser el propio editor de Visual Basic quien establezca la sangría automática. Si introducimos sangrías manualmente, el primer nivel de sangrado corresponderá a una primera pulsación del tabulador, el segundo nivel a dos pulsaciones y así sucesivamente...

A través del menú Opciones, Herramientas... (en algunas versiones hay que entrar además en Editor de texto, Basic, Tabulaciones) Visual Basic nos da la posibilidad de configurar el sangrado que deseamos a través de dos opciones:

1. **Sangría automática:** cuando incluyamos una tabulación en la primera línea de código todas las líneas siguientes comenzarán en esa posición de tabulación.
2. **Ancho de tabulación (tamaño de tabulación):** establece el ancho de la tabulación, ancho que puede oscilar entre 1 y 32 espacios, con un valor predeterminado de 4 espacios. Podemos establecer el valor que mejor se adapte a nuestras necesidades o gustos estéticos.

Una posibilidad que puede resultar interesante es la de usar un símbolo de prolongación de línea cuando tenemos una línea muy larga que no queremos separar a través de un símbolo de dos puntos. Para ello usamos el símbolo de concatenación de línea, que es un guión bajo ( \_ ). Dicho símbolo indicará que el texto que aparece en dos líneas, a efectos de ejecución del programa será tratado como si de una línea se tratara. Veamos el siguiente ejemplo:

*Código (versiones VB menos recientes):*

*Código (versiones VB más recientes):*

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim A%, B%, C%, D%

Private Sub Form_Load()
A = 3
B = 2
C = A * B
D = C ^ 2
MsgBox ("Los valores de las variables que intervienen en
el proceso, tal y como " _
& "han sido definidas en el manual del programa son " _
& " A =" & A & " B =" & B & " C =" & C & " D =" & D)
End Sub
```

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim A, B, C, D%

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        A = 3
        B = 2
        C = A * B
        D = C ^ 2
        MsgBox("Los valores de las variables
que intervienen en el proceso, tal y como " _
& "han sido definidas en el manual
del programa son " _
& " A =" & A & " B =" & B & " C =" &
C & " D =" & D)
    End Sub
End Class
```

Introducimos un símbolo de guión bajo cada vez que queramos continuar una línea. Hemos dividido una línea larga en tres renglones de longitud media. Los motivos para ello pueden ser varios, como estimarlo mejor para la ejecución del programa, o simplemente evitar que en el visor del editor nos quede la línea cortada y que podamos verla en toda su longitud. Prueba a escribir la línea dividiéndola en cuatro a través del guión bajo o dejándola completa sin divisiones. ¿No te parece que "el efecto" visual es muy distinto?

**Próxima entrega: CU00317A**

**Acceso al curso completo en aprenderaprogramar.com** --> Cursos, o en la dirección siguiente:  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=37&Itemid=61](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61)