

```

Module Module1
    Const fmax = 23, cmax = 79 ' fmax=nºde filas máximo, cmax=nº de columnas max, importante que sean impares
    Dim m(cmax, fmax) As Char ' m es una matriz o tablero de caracteres donde iremos haciendo el laberinto
    Dim muro As Char = "#" 'exterior , aquí definimos que caracteres forman el laberinto
    Dim camino As Char = "+" ' camino
    Dim obstaculo As Char = "@" ' obstaculo
    Dim ci, fi, cf, ff As Integer 'posiciones inicial y final del laberinto
    Dim generadoraleatorio As Random = New Random ' Declararlo siempre como variable global, si no, cada vez que se genera
    ' un nº aleatorio se genera el mismo
    -----
    Sub Main()

        Randomize()

        Iniciar_matriz()

        Poner_limites()

        Poner_obstaculos()

        fi = 2 : ci = 2 'Poner Posicion inicial (debe ser par)
        ff = 20 : cf = 70 'Poner Posicion final (debe ser par)

        genera(ci, fi) ' Iniciar algoritmo recursivo generador del laberinto desde la posición inicial

        visualizar() : esperar-tecla() ' Ver resultado

    End Sub
    -----
    Sub Iniciar_matriz() 'iniciar matriz= ponerlo todo como espacios en blanco
        Dim c, f As Integer
        For f = 0 To fmax - 1
            For c = 0 To cmax - 1
                m(c, f) = " "
            Next
        Next
    End Sub
    -----
    Sub Poner_limites() 'AQUÍ Los limites son rectangulares pero pueden tener cualquier forma
        Dim c, f As Integer
        ' limite izquierdo y derecho
        For f = 0 To fmax - 1
            m(0, f) = muro
            m(cmax - 1, f) = muro
        Next f
        ' limite superior e inferior
        For c = 0 To cmax - 1
            m(c, 0) = muro
            m(c, fmax - 1) = muro
        Next c
    End Sub
    -----
    Sub Poner_obstaculos() ' en este caso es un único obstáculo rectangular, pero pueden ser varios y de la forma que quieras
        Dim cobs, fobs, ancho, alto, i, j As Integer

        fobs = 6 : cobs = 40 ' posición par
        ancho = 3 : alto = 11 ' tamaño impar
        For i = 0 To ancho - 1
            For j = 0 To alto - 1
                m(cobs + i, fobs + j) = obstaculo
            Next j
        Next i
    End Sub
    -----
    Sub visualizar()
        For f = 0 To fmax - 1
            For c = 0 To cmax - 1
                System.Console.SetCursorPosition(c, f)
                System.Console.Write(m(c, f))
            Next c
        Next f
        System.Console.SetCursorPosition(ci, fi) ' resaltar origen y final
        System.Console.ForegroundColor = ConsoleColor.Yellow
        System.Console.Write(camino)
        System.Console.SetCursorPosition(cf, ff)
        System.Console.ForegroundColor = ConsoleColor.Red
        System.Console.Write(camino)
        System.Console.ResetColor()
    End Sub
    -----
    Function calculanumerodecaminos(ByVal c As Integer, ByVal f As Integer)
        Dim n As Integer = 0 ' n almacena el nº de caminos posibles desde (c,f) en un instante
        If m(c, f + 2) = " " Then n = n + 1 'si hay camino hacia abajo incremento n
    End Function

```





```

If m(c, f - 2) = " " Then n = n + 1 'si hay camino hacia arriba incremento n
If m(c - 2, f) = " " Then n = n + 1 'si hay camino hacia izquierda incremento n
If m(c + 2, f) = " " Then n = n + 1 'si hay camino hacia derecha incremento n
Return n ' devuelvo n, que es el n° de caminos libres en la posición (c,f)
End Function
-----
Sub genera(ByVal c, ByVal f)
Dim n, numeroaleatorio, opcion As Integer
'1.- Pinta la casilla que me envian
m(c, f) = camino
visualizar() ' Esto en gris no es necesario, solo sirve para ir viendo como se va haciendo poco a poco pulsando ENTER
n = calculanumerodecaminos(c, f) 'en n guardamos el numero de caminos libres en (c,f)
System.Console.SetCursorPosition(0, 24)
System.Console.Write("n=" & n)
System.Console.Write(" f=" & f & " c=" & c & " ")
System.Console.ReadLine()
'2.- mientras haya caminos libres y no este en la meta
While (n <> 0) And Not ((c = cf) And (f = ff))
'2.1.- elegir camino entre los posibles
numeroaleatorio = generadoraleatorio.Next(1, n + 1) 'elijo un n° entre 1 y el numero de caminos libres
System.Console.SetCursorPosition(20, 24)
System.Console.Write("alea=" & numeroaleatorio)
opcion = 1 ' Al menos hay un camino libre, si no, no estaría aquí, OPCION variara entre 1 y el n° de caminos libres
If m(c, f + 2) = " " Then ' Hay camino libre hacia abajo
If opcion = numeroaleatorio Then ' Si esta opción es 1ª elegida...
m(c, f + 1) = camino
genera(c, f + 2) '2.2.- LLAMO RECURSIVAMENTE A GENERA desde la NUEVA POSICION ELEGIDA.
End If ' Se avanza de 2 en 2 para dejar sitio a las paredes entre dos caminos
opcion = opcion + 1 ' Si estoy aquí es porque había camino libre en esta dirección,
' incremento para poder probar luego con otras
End If
If m(c, f - 2) = " " Then 'camino libre arriba
If opcion = numeroaleatorio Then
m(c, f - 1) = camino
genera(c, f - 2)
End If
opcion = opcion + 1
End If
If m(c - 2, f) = " " Then 'camino libre izquierda
If opcion = numeroaleatorio Then
m(c - 1, f) = camino
genera(c - 2, f)
End If
opcion = opcion + 1
End If
If m(c + 2, f) = " " Then 'camino libre derecha
If opcion = numeroaleatorio Then
m(c + 1, f) = camino
genera(c + 2, f)
End If
opcion = opcion + 1
End If
'2.3.- Actualizar el n° de caminos libres en (c,f)
n = calculanumerodecaminos(c, f)
End While
End Sub
-----
Sub esperaratecla()
Dim ch As Char
System.Console.SetCursorPosition(60, 24)
System.Console.Write("s para salir")
Do
ch = System.Console.ReadKey.KeyChar
Loop Until ch = "s"
End Sub
-----
End Module

```

### ALGORITMO GENERA (c,f)

- 1.- Pinta la casilla (c,f)
- 2.- Mientras (haya caminos libres) y no (hayamos llegado a la meta)
  - 2.1.- Elegir\_un\_camino de los posibles
  - 2.2.- Llamar a la subrutina recursiva GENERA desde la nueva posición según el camino elegido antes
  - 2.3.- Actualizar numero de caminos libres en (c,f)

#### Algoritmo 2.1: Elegir\_un\_camino

- 2.1.1.- Contar el n° de posibles caminos
- 2.1.2.- Elegir un numero aleatorio entre 1 y el n° de caminos
- 2.1.3.- Probar con todas las direcciones ( por alguna o algunas habrá camino )
  - 2.1.3.1.- Pruebo una dirección y si esta libre le asigno un n° de opcion
  - 2.1.3.2.- Si el n° de opción coincide con el n° aleatorio será el camino elegido