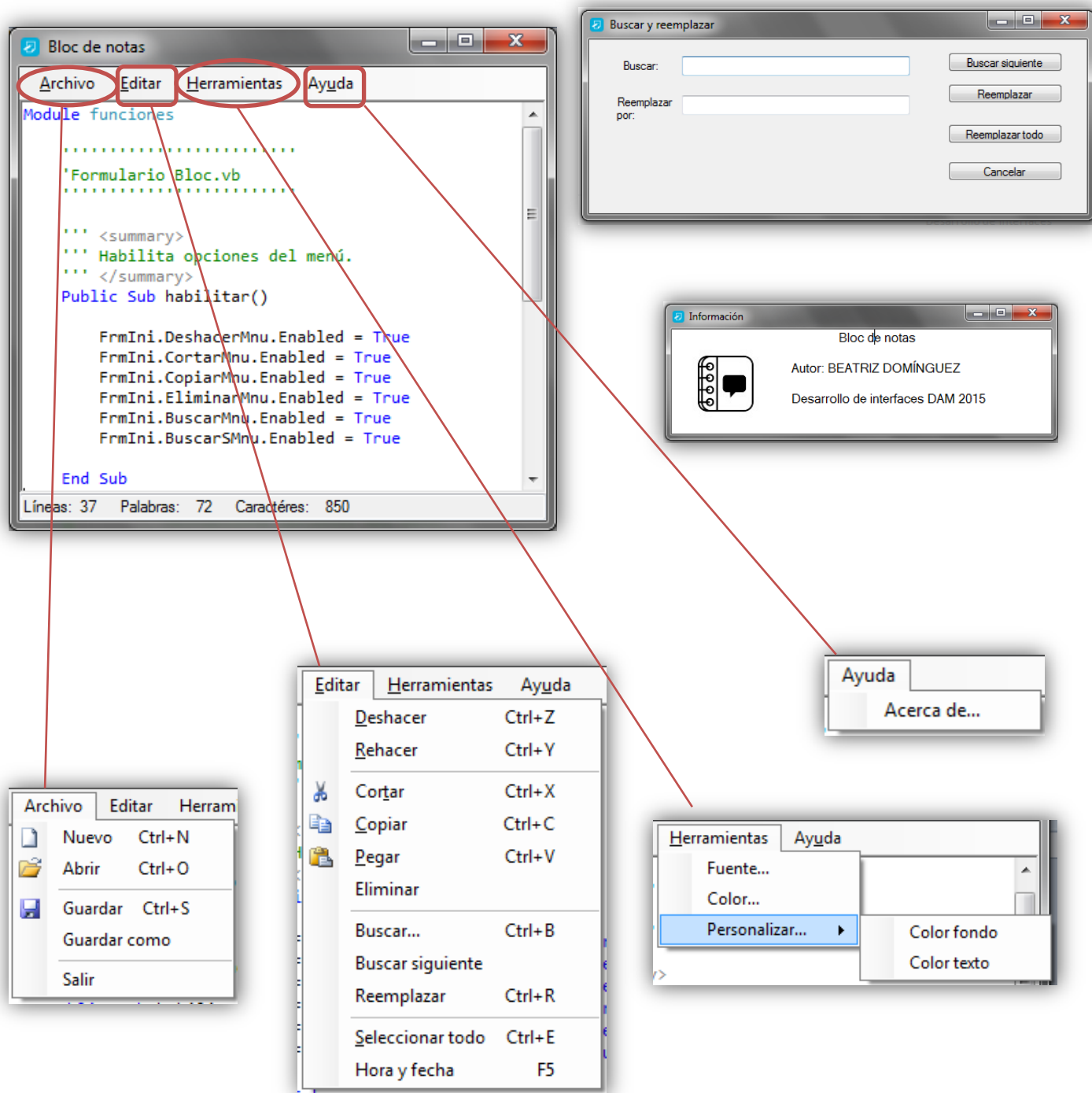


Bloc de notas

Trabajo 02

Formulario



Modificaciones

| Control | Propiedad modificada |
|---|--|
| Frmlni (Windows Form) | Text: Bloc de notas Icon: <i>bloc.ico</i> |
| Mnu (MenuStrip) | Insertar elementos estándar Items: <u>Archivo</u> : Nuevo/ Abrir/ Guardar/ Guardar como / Salir <u>Editar</u> : Deshacer/ Rehacer/ Cortar/ Copiar/ Pegar/ Eliminar/ Buscar.../ Buscar siguiente/ Reemplazar/ Seleccionar todo/ Hora y fecha <u>Herramientas</u> : Fuente.../ Color.../ Personalizar-> Color fondo/ Color texto <u>Ayuda</u> : Acerca de... |
| RTBText (RichTextBox) | Dock: Fill Multiline: True |
| txtCont (TextBox) | Dock: Bottom ReadOnly: True |
| OFDialog (OpenFileDialog) SFDialog (SaveFileDialog) | |
| FntDialog1 (FontDialog) ColorDialog1 (ColorDialog) | |
| BuscarYReemplazar (Windows Form) | Text: Buscar y reemplazar FormBorderStyle: FixedDialog Icon: <i>bloc.ico</i> |
| LblBuscar (Label) LblReplace (Label) | Text: Buscar: Text: Reemplazar por: |
| TxtBuscar (TextBox) TxtReplace (TextBox) | |
| BtnBuscar (Button) BtnReplace (Button) BtnReplaceAll (Button) BtnCancelar (Button) | Text: Buscar siguiente Text: Reemplazar Text: Reemplazar todo Text: Cancelar |
| Info (Windows Form) | Text: Información FormBorderStyle: FixedDialog Icon: <i>bloc.ico</i> |
| RTBInfo (RichTextBox) | ReadOnly: True BackColor: White Dock: Right |
| Pic (PictureBox) | Dock: Left Image: <i>icono.jpg</i> |

Código

Fichero: *Bloc.vb*

```
Imports System.IO
Imports System.Text.RegularExpressions
```

```
''' <summary>
''' Editor de texto
''' </summary>
''' <Autor>
''' Beatriz Domínguez
''' </Autor>
```

```
Public Class FrmIni
```

```
#Region "Variables"
```

```
    'nombre del fichero
    Dim fichero As String = ""
    'flojos de entrada y salida
    Dim sw As StreamWriter
    Dim sr As StreamReader
    'boolean para saber si se han producido cambios o no
    Dim cambios As Boolean = False
```

```
#End Region
```

```
#Region "Archivo"
```

```
    ''' <summary>
    ''' Creamos un nuevo archivo.
    ''' </summary>
    ''' <remarks>
    ''' Si hay cambios sin guardar se da la opción de guardar antes de crear el nuevo
    archivo.
    ''' </remarks>
    Private Sub Nuevo(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
NuevoMnu.Click
```

```
        Dim Result As DialogResult
```

```
        'Si hay cambios a guardar.
        If cambios = True Then
```

```
            'Mostramos el mensaje y guardamos el resultado.
            Result = MessageBox.Show("¿Desea guardar el archivo?", "Nuevo",
MessageBoxButtons.YesNoCancel)
```

```
            'NO
            If Result = 7 Then
```

```
                'Reiniciamos todas las variables.
                RTBText.Clear()
                fichero = ""
                deshabilitar()
                cambios = False
```

```
            'YES
            ElseIf Result = 6 Then
```

```
                'Guardamos el archivo.
                GuardarComo(sender, e)
```

```
                'Reiniciamos todas las variables.
```

```

        RTBText.Clear()
        fichero = ""
        deshabilitar()
        cambios = False

        'CANCEL
    ElseIf Result = 2 Then

        'no hace nada. Volvemos al formulario.
    End If

```

```
End If
```

```
End Sub
```

```

''' <summary>
''' Abrimos un archivo existente.
''' </summary>

```

```
Private Sub Abrir(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AbrirMnu.Click
```

```

    'Creamos el filtro.
    OFDialog.Filter = "Documento de texto (*.txt)|*.txt| Todos los archivos(*.*)|*.*"

    'Seleccionaos el fichero.
    If OFDialog.ShowDialog() = Windows.Forms.DialogResult.OK Then

        'Guardamos el nombre del fichero.
        fichero = OFDialog.FileName

        'Abrimos el fichero.
        sr = File.OpenText(OFDialog.FileName)
        'Cargamos el texto del fichero.
        RTBText.Text = sr.ReadToEnd
        'cerramos el fichero
        sr.Close()
    End If

```

```
End Sub
```

```

''' <summary>
''' Guardamos el fichero.
''' </summary>

```

```
Private Sub Guardar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
GuardarMnu.Click
```

```

    'Guardamos el fichero si tiene nombre (ya ha sido guardado o abierto).
    If fichero <> "" Then

        'Abrimos el fichero.
        sw = File.CreateText(fichero + ".txt")

        'Escribimos la informacion en el fichero.
        sw.Write(RTBText.Text)
        'Vaciar memoria.
        sw.Flush()
        'Cerrar el fichero.
        sw.Close()

        'Si no tiene nombre llamamos al método GuardarComo().
    Else

```

```
        Me.GuardarComo(sender, e)
```

```
End If
```

```
'No hay nuevos cambios que guardar.  
cambios = False  
End Sub
```

```
''' <summary>  
''' Guardamos el fichero definiendo nombre y ubicación.  
''' </summary>  
Private Sub GuardarComo(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles GuardarCMnu.Click  
  
    'Definimos un filtro.  
    SFDialog.Filter = "Documento de texto (*.txt)|*.txt| Todos los archivos(*.*)|*.*"  
  
    'Abrimos el cuadro de diálogo.  
    If SFDialog.ShowDialog() = Windows.Forms.DialogResult.OK Then  
  
        'Guardamos el nombre del fichero.  
        fichero = SFDialog.FileName  
  
        'Abrimos el fichero.  
        sw = File.CreateText(SFDialog.FileName)  
        'Escribimos la informacion en el fichero.  
        sw.Write(RTBText.Text)  
        'Vaciar memoria.  
        sw.Flush()  
        'Cerrar  
        sw.Close()  
  
    End If  
  
    'No hay nuevos cambios que guardar.  
    cambios = False
```

```
End Sub
```

```
''' <summary>  
''' Salimos del programa.  
''' </summary>  
Private Sub Salir(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles  
SalirMnu.Click  
  
    'Cerramos el formulario.  
    Me.Close()
```

```
End Sub
```

```
#End Region
```

```
#Region "Editar"
```

```
''' <summary>  
''' Deshace la última acción realizada.  
''' </summary>
```

```
Private Sub Deshacer(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles  
DeshacerMnu.Click  
  
    'Determina si se puede deshacer la última acción.  
    If RTBText.CanUndo = True Then  
        ' Deshace la acción.  
        RTBText.Undo()
```

```
    End If  
End Sub
```

```
''' <summary>  
''' Rehace la última acción realizada.  
''' </summary>
```

```

Private Sub RehacerMnu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RehacerMnu.Click

    'Determina si se puede rehacer la última acción.
    If RTBText.CanRedo = True Then

        'rehace la última acción.
        RTBText.Redo()

    End If

End Sub

```

```

''' <summary>
''' Corta el texto seleccionado.
''' </summary>
Private Sub Cortar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CortarMnu.Click

    'Si hay texto seleccionado lo corta
    If RTBText.SelectionLength > 0 Then
        RTBText.Cut()
    End If
End Sub

```

```

''' <summary>
''' Copia el texto seleccionado.
''' </summary>
Private Sub Copiar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CopiarMnu.Click

    'si hay texto seleccionado lo copia
    If RTBText.SelectionLength > 0 Then
        RTBText.Copy()
    End If
End Sub

```

```

''' <summary>
''' Pega el texto que hay en el portapapeles en la posición del cursos.
''' </summary>
Private Sub Pegar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PegarMnu.Click

    'Determina si hay texto en el portapapeles para pegar.
    If Clipboard.GetDataObject().GetDataPresent(DataFormats.Text) = True Then
        'Determina si hay texto seleccionado.
        If RTBText.SelectionLength > 0 Then

            'Preguntar al usuario si quieres pegar sobre el texto seleccionado.
            If MessageBox.Show("Do you want to paste over current selection?", _
                "Cut Example", MessageBoxButtons.YesNo) = DialogResult.No Then
                'En caso negativo, mover el cursor después de la selección.
                RTBText.SelectionStart = RTBText.SelectionStart + _
                    RTBText.SelectionLength
            End If
        End If
        'Pegar el texto guardado en el portapapeles.
        RTBText.Paste()
    End If

End Sub

```

```

''' <summary>
''' Elimina el texto seleccionado
''' </summary>

```

```

Private Sub Eliminar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
EliminarMnu.Click
    'si hay texto seleccionado lo elimina.
    If RTBText.SelectionLength > 0 Then
        RTBText.SelectedText = ""
    End If
End Sub

```

```

''' <summary>
''' Abre el formulario para buscar un texto.
''' </summary>
Private Sub Buscar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BuscarMnu.Click
    BuscarYReemplazar.Show()
End Sub

```

```

''' <summary>
''' Abrimos el formulario de buscar.
''' </summary>
Private Sub BuscarSiguiente(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BuscarSMnu.Click
    BuscarYReemplazar.Show()
End Sub

```

```

''' <summary>
''' Abre el formulario para reemplazar texto.
''' </summary>
Private Sub Reemplazar(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ReemplazarMnu.Click
    BuscarYReemplazar.Show()
End Sub

```

```

''' <summary>
''' Cambia la fuente del texto seleccionado o de todo el documento.
''' </summary>
Private Sub Fuente(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
FuenteMnu.Click

    'Abrimos el diálogo y seleccionadmos 'Aceptar'.
    If FntDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        'Si hay texto seleccionado aplica los cambios a ese texto.
        If RTBText.SelectedText.Count > 0 Then
            RTBText.SelectionFont = FntDialog1.Font

            'Si no hay texto seleccionado aplica los cambios a todo el texto del
documento.
        Else
            RTBText.Font = FntDialog1.Font
        End If
    End If
End Sub

```

```

''' <summary>
''' Cambia el color del texto seleccionado o de todo el documento.
''' </summary>
Private Sub Color(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ColorMnu.Click

    'Abrimos el diálogo y seleccionadmos 'Aceptar'.
    If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        'Si hay texto seleccionado aplica los cambios a ese texto.
        If RTBText.SelectedText.Count > 0 Then

```



```

        RTBText.SelectionColor = ColorDialog1.Color

        'Si no hay texto seleccionado aplica los cambios a todo el texto del
documento.
    Else

        RTBText.ForeColor = ColorDialog1.Color

    End If

End If
End Sub

```

```

''' <summary>
''' Seleccionamos todo el texto del documento.
''' </summary>
Private Sub SeleccionarTodo(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles SeleccMnu.Click

    'Seleccionamo el texto
    RTBText.SelectAll()

End Sub

```

```

''' <summary>
''' Añadimos el la fecha y hora actual del sistema en el texto en la posición del
cursor.
''' </summary>
''' <remarks>
''' Formato: 10/11/2014 21:34
''' </remarks>
Private Sub HoraYFechaMnu_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles HoraYFechaMnu.Click

    'Variable auxiliar para guardar lo que hay en el portapapeles al llamar al método.
    Dim portapapeles As String

    'Guardamos el portapapeles en la variable, y definimos la hora y fecha en el
portapapeles.
    portapapeles = Clipboard.GetText
    Clipboard.SetText(DateTime.Now.ToString("g"))

    'Determina si hay texto seleccionado.
    If RTBText.SelectionLength > 0 Then
        'Preguntamos al usuario si quiere pegar sobre el texto seleccionado.
        If MessageBox.Show("Do you want to paste over current selection?", _
            "Cut Example", MessageBoxButtons.YesNo) = DialogResult.No Then

            'Movemos el cursor después del texto seleccionado.
            RTBText.SelectionStart = RTBText.SelectionStart + _
                RTBText.SelectionLength
        End If
    End If

    'Pegamos el portapapeles (hora y fecha).
    RTBText.Paste()

    'Volvemos a definir el portapapeles como al principio con la variable auxiliar.
    Clipboard.SetText(portapapeles)

End Sub

```

```

#End Region

```

```

''' <summary>
''' Muestra un cuadro informativo.

```

```

''' </summary>
Private Sub AcercaDe(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
AcercaMnu.Click

    Info.Show()

End Sub

```

```

''' <summary>
''' Método que se ejecuta al cerrarse el formulario.
''' </summary>
Private Sub FrmIni_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

    'Variable para el resultado del dialogo de mensaje.
    Dim Result As DialogResult

    'Si se han producido cambios se muestra un mensaje antes de salir con la posibilidad
de guardar el archivo o volver a el.
    If cambios = True Then

        'Mostramos el mensaje y guardamos el resultado
        Result = MessageBox.Show("¿Desea guardar los cambios realizados?", "Salir",
MessageBoxButtons.YesNoCancel)

        'NO
        If Result = 7 Then
            'Definimos que no haya cambios que guardar.
            cambios = False
            'Cerramos el formulario.
            Me.Close()

            'YES
            ElseIf Result = 6 Then

                'Guardamos el archivo
                GuardarComo(sender, e)

                'CANCEL
                ElseIf Result = 2 Then
                    'Volvemos al formulario.
                    Dim form As FrmIni
                    form.Show()

                End If

            End If

        End Sub

```

```

''' <summary>
''' Cuenta el número de palabras.
''' </summary>
''' <returns>Número de palabras en el texto</returns>
Public Function CountWords() As Integer
    Dim collection As MatchCollection = Regex.Matches(RTBText.Text, "\S+")
    'Devolvemos el número de palabras.
    Return collection.Count
End Function

```

```

''' <summary>
''' Método que se ejecutará cada vez que se produzca un cambio en el texto. Guardar
cambios y mostrar utilidades.
''' </summary>
Private Sub TextChanged_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RTBText.TextChanged

```

```

        'mostramos el número de líneas, palabras y caracteres que hay
        txtCont.Text = "Líneas: " & RTBText.Lines.Length & " Palabras: " &
CountWords() & " Caracteres: " & RTBText.TextLength

        'habilitamos o deshabilitamos funciones según si haya o no texto
        If RTBText.Text <> "" Then
            habilitar()

            'Definimos que se han producido cambios a guardar.
            cambios = True

        Else
            deshabilitar()

            'Definimos que los cambios realizados no se deben guardar.
            cambios = False
        End If

    End Sub

```

```

#Region "Apariencia"

''' <summary>
''' Cambiamos el color de fondo de los menus del editor.
''' </summary>
Private Sub ColorFondo(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ColorFondoMnu.Click

    'Mostramos el diálogo y hacemos clic en "Aceptar".
    If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        'Cambiamos el color de fondo por el seleccionado.
        Mnu.BackColor = ColorDialog1.Color
        txtCont.BackColor = ColorDialog1.Color

    End If

End Sub

```

```

''' <summary>
''' Cambiamos el color del texto de los menus del editor.
''' </summary>
Private Sub ColorTexto(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ColorTextoToolStripMenuItem.Click

    'Mostramos el diálogo y hacemos clic en "Aceptar".
    If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        'Cambiamos el color de fondo por el seleccionado.
        Mnu.ForeColor = ColorDialog1.Color
        txtCont.ForeColor = ColorDialog1.Color

    End If
End Sub
#End Region

End Class

```

Fichero: BuscarYReemplazar.vb

```

''' <summary>
''' Editor de texto. Formulario de búsqueda y reemplazo

```

```
''' </summary>
''' <Autor>
''' Beatriz Domínguez
''' </Autor>
```

Public Class BuscarYReemplazar

```
'variables para la busqueda
Dim lastIndex As Integer
Dim strSearchTerm As Integer
Dim index As Integer
Dim strLastSearch As Integer
```

```
''' <summary>
''' cierra el formulario
''' </summary>
```

```
Private Sub BtnCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnCancelar.Click
    Me.Close()
End Sub
```

```
''' <summary>
''' busca en el texto la siguiente instancia de una cadena
''' </summary>
```

```
Private Sub BuscarS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnBuscar.Click
```

```
'Buscamos la primera ocurrencia de una cadena y las siguiente hasta alcanzar el fin
del documento
lastIndex = FrmIni.RTBText.Text.LastIndexOf(TxtBuscar.Text) 'Indica la posicion de
la ultima ocurrencia de una cadena
```

```
If lastIndex = 0 Then 'Si la ultima ocurrencia en 0 quiere decir que esta al
comienzo del documento
    strSearchTerm = 0 'Establecemos a 0
```

```
ElseIf lastIndex = -1 Then 'Si no existe la ultima ocurrencia
```

```
    If FrmIni.RTBText.Text = "" Then 'Si el txtContenido está vacío
        MsgBox("No se encontró '" & TxtBuscar.Text & "'.", 48, "Atención")
```

```
    Else
```

```
        If FrmIni.RTBText.Find(TxtBuscar.Text, 0, FrmIni.RTBText.Text.Length,
RichTextBoxFinds.None) = -1 Then 'Si no está vacío buscamos de nuevo
            MsgBox("No se encontró '" & TxtBuscar.Text & "'.", 48, "Atención")
```

```
        Else
```

```
            strSearchTerm = FrmIni.RTBText.Find(TxtBuscar.Text, 0,
FrmIni.RTBText.Text.Length, RichTextBoxFinds.None) 'Si se encuentra establecemos
strSearchTerm con el índice de inicio de la ocurrencia.
```

```
        End If
```

```
    End If
```

```
    Else
```

```
        strSearchTerm = FrmIni.RTBText.Find(TxtBuscar.Text, index, lastIndex,
RichTextBoxFinds.None) 'Si nada de lo anterior de cumple establecemos strSearchTerm al
índice de la ocurrencia
```

```
    End If
```

```
If strSearchTerm = -1 Then
```

```
    strLastSearch = FrmIni.RTBText.Text.LastIndexOf(TxtBuscar.Text) 'Si se encuentra
pero es la última, repetimos el bucle
```

```
    FrmIni.Focus()
```

```
    FrmIni.RTBText.SelectionStart = strLastSearch
```

```
    FrmIni.RTBText.SelectionLength = TxtBuscar.Text.Length
```

```
    index = 0
```

```
Else 'Si se encuentra y no es la última, establecer index a partir de la declaración
Find para evitar la aparición anterior
```

```
    FrmIni.Focus()
```

```

        FrmIni.RTBText.SelectionStart = strSearchTerm
        FrmIni.RTBText.SelectionLength = TxtBuscar.Text.Length
        index = strSearchTerm + 1
    End If

    ' End If
End Sub

```

```

''' <summary>
''' Reemplaza la primera instancia de una cadena por otra cadena dada
''' </summary>
Private Sub BtnReplace_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnReplace.Click

    'si no encuentra la cadena a reemplazar muestra un mensaje por pantalla
    If FrmIni.RTBText.Text.IndexOf(TxtBuscar.Text) = -1 Then
        MessageBox.Show("No se ha encontrado el texto a reemplazar", "Reemplazar",
        MessageBoxButtons.OK)

    Else
        'reemplaza la primera instancia de la cadena
        FrmIni.RTBText.Text = Replace(FrmIni.RTBText.Text, TxtBuscar.Text,
        TxtReplace.Text, , 1)
        FrmIni.Focus()

    End If
End Sub

```

```

''' <summary>
''' reemplaza todas las instancias de una cadena por otra dada
''' </summary>
Private Sub BtnReplaceAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnReplaceAll.Click

    'si no encuentra la cadena a reemplazar muestra un mensaje por pantalla
    If FrmIni.RTBText.Text.IndexOf(TxtBuscar.Text) = -1 Then
        MessageBox.Show("No se ha encontrado el texto a reemplazar", "Reemplazar",
        MessageBoxButtons.OK)

    Else

        'reemplaza las cadenas
        FrmIni.RTBText.Text = Replace(FrmIni.RTBText.Text, TxtBuscar.Text,
        TxtReplace.Text)
        FrmIni.Focus()

    End If

End Sub

```

End Class

Archivo: funciones.vb

Module funciones

```

.....
'Formulario Bloc.vb
.....

''' <summary>
''' Habilita opciones del menú.
''' </summary>
Public Sub habilitar()

```

```

    FrmIni.DeshacerMnu.Enabled = True

```

```
FrmIni.CortarMnu.Enabled = True
FrmIni.CopiarMnu.Enabled = True
FrmIni.EliminarMnu.Enabled = True
FrmIni.BuscarMnu.Enabled = True
FrmIni.BuscarSMnu.Enabled = True
```

```
End Sub
```

```
''' <summary>
''' Deshabilita opciones del menú.
''' </summary>
```

```
Public Sub deshabilitar()
```

```
    FrmIni.DeshacerMnu.Enabled = False
    FrmIni.CortarMnu.Enabled = False
    FrmIni.CopiarMnu.Enabled = False
    FrmIni.EliminarMnu.Enabled = False
    FrmIni.BuscarMnu.Enabled = False
    FrmIni.BuscarSMnu.Enabled = False
```

```
End Sub
```

```
End Module
```