



Universitat
Autònoma
de Barcelona



**CODI FONT DE L'APLICACIÓ
ASSOCIADA AL PROJECTE**

**SOFTWARE DE MONITORITZACIÓ
DE DADES REBUDES EN UN GPS**

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Alejandro Comín Gimeno
i dirigit per
Joan Oliver Malagelada
Bellaterra, 12 de Juny de 2006

ÍNDEX

1. FORMULARIS	1
1.1. frmPrincipal.vb	1
1.2. frmPropietatsCalibratge.vb	18
1.3. frmPuntsCalibratge.vb	19
1.4. frmConfigCom.vb	24
1.5. frmCanviCoord.vb	26
2. CLASSES	27
2.1. elipsoide.vb	27
2.2. canvidatum.vb	27
2.3. costats.vb	32
2.4. NmeaInterpreter.vb	37

CODI FONT DE L'APLICACIÓ

1. FORMULARIS

1.1. frmPrincipal.vb

```
Imports PFC.canvidatum
Imports PFC.elipsoide
Imports PFC.costats
Imports PFC.mdlVarsCompartides
Imports PFC.NmeaInterpreter

Public Class frmPrincipal

    Inherits System.Windows.Forms.Form

    '-----
    '----- TIMERS -----
    '-----

    'Timer encarregat dels punts de calibratge
    Private timerPunts As New System.Timers.Timer(50)

    'Timer encarregat d'actualitzar les coordenades pantalla
    Private timerInfo As New System.Timers.Timer(50)

    'Timer que actualitza la informació UTM per pantalla
    Private timerUTM As New System.Timers.Timer(50)

    '-----
    '----- TRACKS -----
    '-----

    'Array que conté els punts de track en coordenades pantalla
    Private arrayTrack(4000) As Point

    'Número de punts totals del track carregat
    Private numPuntTrack As Integer = 0

    'Variable que controla si el track s'està mostrant o no
    Private mostrarTrack As Boolean = False

    '-----
    '----- COMUNICACIÓ GPS -----
    '-----

    'Variable del port COM de comunicació amb el GPS
    Friend WithEvents pSerie As New IO.Ports.SerialPort()

    'Variable que parseja les comandes NMEA rebudes
    Private cNMEA As New NmeaInterpreter

    'Canvi de datum necessari entre les dades rebudes del GPS i el datum del mapa
    Private canviDatumGPS As New canviDatum

    'Proporcions dels costats d'acord amb el datum original
    Private costatsMapaDatumCalibratge As New costats

    'Variable que ens indica la posició actual en coordenades pantalla del GPS
    Private coordPantallaGPS As New Point
```

```

'-----
'----- ALTRES -----
'-----

'Control de la posició de click per al desplaçament de la imatge (pan)
Private posicioClickInicial As New Point

'Variable que controla les proporcions coord Pantalla - coord UTM
Private costatsMapa As New costats

#Region "Menú Arxiu"

Private Sub menuCarregar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuCarregar.Click

    'Crear un diàleg d'apertura del tipus d'imatge
    Dim dlgObrirMapa As New OpenFileDialog

    'Extensió d'arxius que es mostren
    dlgObrirMapa.Filter = "Arxius d'imatges JPG (*.jpg)|*.jpg|Tots els arxius
d'imatges|*.gif;*.jpg;*.bmp;*.png"

    'Es comprova si s'apreta el botó d'Acceptar
    If dlgObrirMapa.ShowDialog() = Windows.Forms.DialogResult.OK Then

        'Sobre l'arxiu i es modifica la mida del pictureBox al formulari
        imgMapa.Image = Image.FromFile(dlgObrirMapa.FileName)
        imgMapa.Width = imgMapa.Image.Width
        imgMapa.Height = imgMapa.Image.Height

        'Es guarda la mida original de la imatge, per no tenir problemes en calibrar
        midaMapa = imgMapa.Size
        factorEscala = 1
        comboZoom.Text = "100%"

        'Actualitzar el menú amb les noves opcions disponibles
        menuTancar.Enabled = True
        menuCalibratge.Enabled = True
        menuCarregar.Enabled = False
        menuZoom.Enabled = True

        'Altres inicialitzacions
        dlgObrirMapa.FileName.Substring(0, dlgObrirMapa.FileName.LastIndexOf("."))
        imgCarregada = True
        numPuntTrack = 0

        'Iniciar els timers d'informació i de punts
        timerInfo.Start()
        timerPunts.Start()

    End If

End Sub

Private Sub menuTancar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuTancar.Click

    Dim msg As MsgBoxResult

    msg = MsgBox("Estàs a punt de tancar el mapa actual. Es perdran tots els canvis
no salvats." & Chr(13) & Chr(13) & _
        "Segur que vols continuar?", 68, "Atenció")

    'Si s'ha premut el botó del sí tornem a l'estat inicial
    If msg = MsgBoxResult.Yes Then

        'De la imatge
        imgCarregada = False
        imgMapa.Width = 50
        imgMapa.Height = 50
        imgMapa.Image = Nothing

    End If

End Sub

```

```

'Dels menús
menuTancar.Enabled = False
menuCalibratge.Enabled = False
menuDatum.Enabled = False
menuCarregar.Enabled = True
menuZoom.Enabled = False
menuTrack.Enabled = False
menuCOM.Enabled = False

'De les propietats de calibratge
datumCalibratge = ""
mdlVarsCompartides.datumActual = ""
projeccioCalibratge = ""
puntUltimClick = Point.Empty
numPuntsCalibratge = 0
puntsCalibratge = False
jaCalibrat = False
arrayPuntsCalibratge = Nothing
arrayTrack(4000) = Nothing

'De les comunicacions
aturarCOM_Click(Nothing, Nothing)

'De la informació a pantalla
coordX.Text = ""
coordY.Text = ""
datum.Text = ""
projeccio.Text = ""
punts.Text = ""
easting.Text = ""
northing.Text = ""
zona.Text = ""
hemisferi.Text = ""
llegitCOM.Text = ""
datumActual.Text = ""
comboZoom.Text = "100%"

End If

End Sub

Private Sub menuSortir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuSortir.Click

    Dim msg As MsgBoxResult

    msg = MsgBox("Estàs a punt de sortir del programa. Es perdran tots els canvis no
salvats." & Chr(13) & Chr(13) & _
        "Segur que vols sortir?", 68, "Atenció")

    If msg = MsgBoxResult.Yes Then

        Me.Close()

    End If

End Sub

#End Region

#Region "Menú Calibratge"

Private Sub menuCalibrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuCalibrar.Click

    'S'inicia el formulari de propietats del calibratge
    If Calibratge = False Then

        Dim pCalibratge As New frmPropietatsCalibratge
        pCalibratge.Show()

    End If

End Sub

```

```

Private Sub menuCarregarCalibratge_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuCarregarCalibratge.Click

    'Es comprova que no s'estigui mostrant el formulari de punts de calibratge
    If puntsCalibratge = True Then

        MsgBox("Tanca la finestra de calibratge actual abans de carregar-ne un de
nou.", MsgBoxStyle.Exclamation, "Atenció")

    Else

        Dim dlgCarregarPunts As New OpenFileDialog

        'Tipus d'arxius que pot obrir
        dlgCarregarPunts.Filter = "Arxiu de calibratge (*.dat) | *.dat"

        'Es comprova si s'apreta el botó d'Acceptar
        If dlgCarregarPunts.ShowDialog() = Windows.Forms.DialogResult.OK Then

            Try

                Dim arxiu As New System.IO.StreamReader(dlgCarregarPunts.FileName)

                'Es llegeixen les dades d'acord amb el format conegut del fitxer de
calibratge

                With arxiu

                    datumCalibratge = .ReadLine
                    projeccioCalibratge = .ReadLine
                    numPuntsCalibratge = .ReadLine

                    Dim frmPunts As New frmPuntsCalibratge

                    frmPunts.Show()

                    'S'actualitza la variable dels punts de calibratge
                    ReDim arrayPuntsCalibratge(numPuntsCalibratge - 1)

                    'Es guarden els punts de calibratge
                    For i As Integer = 0 To numPuntsCalibratge - 1 Step 1

                        arrayPuntsCalibratge(i).coordPantalla.X = .ReadLine
                        arrayPuntsCalibratge(i).coordPantalla.Y = .ReadLine

                    Next

                    'Es llegeixen les dades de cada un dels punts
                    'i es carreguen al formulari de punts de calibratge
                    frmPunts.zona1.Text = .ReadLine
                    frmPunts.easting1.Text = .ReadLine
                    frmPunts.northing1.Text = .ReadLine
                    frmPunts.ns1.Text = .ReadLine

                    frmPunts.zona2.Text = .ReadLine
                    frmPunts.easting2.Text = .ReadLine
                    frmPunts.northing2.Text = .ReadLine
                    frmPunts.ns2.Text = .ReadLine

                    frmPunts.zona3.Text = .ReadLine
                    frmPunts.easting3.Text = .ReadLine
                    frmPunts.northing3.Text = .ReadLine
                    frmPunts.ns3.Text = .ReadLine

                    'Es tanca el fitxer
                    .Close()

                End With

                Catch ex As Exception

                    MsgBox(ex.ToString)
                    Exit Sub

                End Try

            End Try

        End Try
    End Try

```

```

'S'actualitzen les finestres obertes
puntsCalibratge = True

'S'actualitzen les opcions dels menús
menuTrack.Enabled = True
menuDatum.Enabled = True

'Es comprova el datum i s'activa l'opció de representació
If datumCalibratge = "ED50" Then

    datumED50.Checked = True
    datumWGS84.Checked = False
    datumActual.Text = "ED50"
    mdlVarsCompartides.datumActual = "ED50"

Else

    datumED50.Checked = False
    datumWGS84.Checked = True
    datumActual.Text = "WGS84"
    mdlVarsCompartides.datumActual = "WGS84"

End If

'Es dibuixen els punts carregats a pantalla
imgMapa.Refresh()
redibuixarPunts()

MsgBox("Arxiu de calibratge carregat amb èxit.",
MsgBoxStyle.Information, "Informació")

Else

    'Si s'ha apretat el botó cancel·lar s'informa del fet
    MsgBox("No s'ha seleccionat cap arxiu de calibratge.",
MsgBoxStyle.Exclamation, "Alerta!")

End If

End If

End Sub

Private Sub menuRecalibrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuRecalibrar.Click

    'Es comprova que no es mostri la finestra de calibratge
    If puntsCalibratge = False Then

        'Es comprova si ja ha estat calibrat el mapa
        If jaCalibrat = False Then

            'Si no ho ha estat, ho advertim
            MsgBox("Aquest mapa no ha estat calibrat anteriorment.",
MsgBoxStyle.Information, "Informació")

        Else

            'En cas que sí hagi estat calibrat, s'obre el formulari
            Dim pCalib As New frmPuntsCalibratge

            pCalib.Show()

            'Deixem de mostrar el track
            mostrarTrack = False
            imgMapa.Refresh()

        End If

    Else

        'Si ja està obert un formulari de calibratge s'adverteix
        MsgBox("Si us plau, abans de recalibrar, tanca primer la finestra de
calibratge activa.", MsgBoxStyle.Information, "Informació")
    
```

```

End If

End Sub

Private Sub datumED50_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles datumED50.Click

    'En cas que no estigui seleccionat s'actualitza
    If datumED50.Checked = False Then

        datumED50.Checked = True
        datumWGS84.Checked = False

        'Si el mapa ja ha estat calibrat
        If jaCalibrat = True Then

            'Es fa el canvi de datum de la representació actual del mapa
            datumActual.Text = "ED50"
            calcularNousPunts("WGS84", "ED50")
            costatsMapa.calcularPasos(0)

        End If

    End If

End If

End Sub

Private Sub datumWGS84_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles datumWGS84.Click

    If datumWGS84.Checked = False Then

        datumED50.Checked = False
        datumWGS84.Checked = True

        If jaCalibrat = True Then

            datumActual.Text = "WGS84"
            calcularNousPunts("ED50", "WGS84")
            costatsMapa.calcularPasos(0)

        End If

    End If

End Sub

Private Sub datumActual_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles datumActual.SelectedIndexChanged

    'S'actualitza el datum d'acord amb el seleccionat al formulari principal
    If jaCalibrat = True AndAlso puntsCalibratge = False Then

        Select Case datumActual.Text

            Case "ED50"

                datumED50_Click(Nothing, Nothing)

            Case "WGS84"

                datumWGS84_Click(Nothing, Nothing)

        End Select

    End If

End Sub

Private Sub calcularNousPunts(ByVal dOrigen As String, ByVal dDesti As String)

```

```

Dim canviDatumActual As New canviDatum
Dim coordUTMActual As New coordUTM
Dim cUTMD As New coordUTM

'S'inicia la variable que permet el canvi de datum
canviDatumActual.iniDatum(dOrigen, dDesti)
mdlVarsCompartides.datumActual = dDesti

'Es restableixen les dades originals en cas de seleccionar el datum original
If dOrigen = datumCalibratge Then

    For i As Integer = 0 To numPuntsCalibratge - 1
        arrayPuntsDatumActual(i) = arrayPuntsCalibratge(i)
    Next

End If

'Per a cada coordenada realitzem el canvi de datum
For i As Integer = 0 To numPuntsCalibratge - 1

    With coordUTMActual

        .easting = arrayPuntsDatumActual(i).easting
        .northing = arrayPuntsDatumActual(i).northing
        .hemisferi = arrayPuntsDatumActual(i).hemisferi
        .zona = arrayPuntsDatumActual(i).zona

        'Modificació necessària per a mantenir les dades correctes
        If dOrigen = datumCalibratge Then

            .easting += 1
            .northing += 1

        End If

    End With

    'Es realitza el canvi de datum de la coordenada actual
    cUTMD = canviDatumActual.canviDeDatum(coordUTMActual)

    'S'actualitza el punt actual a l'array de punts de datum
    With cUTMD

        arrayPuntsDatumActual(i).easting = .easting
        arrayPuntsDatumActual(i).northing = .northing
        arrayPuntsDatumActual(i).hemisferi = .hemisferi
        arrayPuntsDatumActual(i).zona = .zona

    End With

Next

End Sub

#End Region

#Region "Menú Zoom"

Private Sub zoom10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles zoom10.Click

    aplicarZoom(0.1)

End Sub

Private Sub zoom25_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles zoom25.Click

    aplicarZoom(0.25)

End Sub

Private Sub zoom50_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles zoom50.Click

```

```

        aplicarZoom(0.5)

    End Sub

    Private Sub zoom100_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles zoom100.Click

        aplicarZoom(1)

    End Sub

    Private Sub zoom200_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles zoom200.Click

        aplicarZoom(2)

    End Sub

    Private Sub zoom500_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles zoom500.Click

        aplicarZoom(5)

    End Sub

    Sub aplicarZoom(ByVal zoom As Double)

        'Es canvia la mida del mapa al formulari principal
        imgMapa.Width = imgMapa.Image.Width * zoom
        imgMapa.Height = imgMapa.Image.Height * zoom

        'S'actualitza el factor d'escala per a punts en pantalla
        factorEscala = zoom

        'S'actualitza el valor al comboBox del formulari principal
        comboZoom.Text = zoom * 100 & "%"

        'Es redibuixa el mapa
        imgMapa.Refresh()
        redibuixarTrack()

    End Sub

    Private Sub comboZoom_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles comboZoom.SelectedIndexChanged

        'S'actualitza el mapa d'acord amb el zoom escollit al comboBox
        If imgCarregada = True Then

            Select Case comboZoom.Text

                Case "10%"
                    zoom10_Click(Nothing, Nothing)

                Case "25%"
                    zoom25_Click(Nothing, Nothing)

                Case "50%"
                    zoom50_Click(Nothing, Nothing)

                Case "100%"
                    zoom100_Click(Nothing, Nothing)

                Case "200%"
                    zoom200_Click(Nothing, Nothing)

                Case "500%"
                    zoom500_Click(Nothing, Nothing)

            End Select
        End If
    End Sub

```

```

        End If

    End Sub

#End Region

#Region "Menú Comunicacions GPS"

    Private Sub iniciarCOM_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles iniciarCOM.Click

        'Es comprova que no està en ús el port COM actual
        'Si ho està, es tanca
        If pSerie.IsOpen = True Then
            pSerie.Close()
        End If

        'S'obre i es comprova que no es produeix cap error
        Try

            pSerie.Open()

        Catch ex As Exception

            MsgBox(ex.ToString)
            Exit Sub

        End Try

        'S'actualitza el menú "Comunicació GPS"
        iniciarCOM.Enabled = False
        aturarCOM.Enabled = True

        'S'inicia la variable que realitza el canvi de datum entre
        'el dispositiu GPS i el datum amb què hem calibrat el mapa
        canviDatumGPS.iniDatum("WGS84", datumCalibratge)

    End Sub

    Private Sub aturarCOM_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles aturarCOM.Click

        'Es tanca el port COM en el cas que estigui obert
        Try

            If pSerie.IsOpen = True Then

                pSerie.Close()

            End If

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        'S'actualitza el menú "Comunicació GPS"
        iniciarCOM.Enabled = True
        aturarCOM.Enabled = False

    End Sub

    Private Sub configPortCom_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles configPortCom.Click

        'Només es mostra el formulari en cas que no estigui ja obert
        If configCOM = False Then

            Dim cCom As New frmConfigCom
            cCom.Show()

        End If
    
```

```

End Sub

'-----
'----- FUNCIONS AUXILIARS -----
'-----

Private Sub DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles pSerie.DataReceived

    'Cada cop que es reben dades pel port COM es crea un thread i les tracta
    Try

        llegendCOM.Invoke(New myDelegate(AddressOf updateTextBox), New Object() {})

    Catch ex As Exception

        ex.ToString()

    End Try

End Sub

Public Delegate Sub myDelegate()

Public Sub updateTextBox()

    With llegendCOM

        Dim llegend As String = pSerie.ReadLine

        'S'actualitza la barra de text amb les dades llegides
        .Text = llegend

        'Es comprova que les dades rebudes són vàlides
        If cNMEA.Parse(llegend) = True Then

            Dim coordUTMGPS As New coordUTM

            'Iniciem el canvi de datum a realitzar per a ubicar en pantalla la
posició

            canviDatumGPS.iniDatum("WGS84", datumCalibratge)

            'Es transformen aquestes dades a coordenades pantalla
            coordUTMGPS = geografiques2UTM(canviDatumGPS,
(cNMEA.coordGeodRebuda.latitud) * (Math.PI / 180), (cNMEA.coordGeodRebuda.longitud) *
(Math.PI / 180))
            UTM2coordPantalla(coordUTMGPS, -1)

        End If

    End With

End Sub

#End Region

#Region "Menú Tracks"

Private Sub carregarTrack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles carregarTrack.Click

    'Es crea un diàleg d'apertura i el tipus d'imatge
    Dim dlgCarregarTrack As New OpenFileDialog

    'Tipus d'arxius que pot obrir (Arxius de datum d'OziExplorer)
    dlgCarregarTrack.Filter = "Arxius de tracks (*.plt)|*.plt|Tots els arxius
d'imatges (*.gif;*.jpg;*.bmp;*.png)"

    'Es comprova si s'apreta el botó d'Acceptar
    If dlgCarregarTrack.ShowDialog() = Windows.Forms.DialogResult.OK Then

        'Es buida el contingut de l'array de punts
    
```

```

For i As Integer = 0 To arrayTrack.Length - 1
    arrayTrack(i) = Point.Empty
Next

'Datum del track
Dim nomDatumTrack As String

'Tractament del fitxer
Dim liniaActual As String
Dim camps() As String
Dim coordenada(2) As String
Dim latitud As Double
Dim longitud As Double

'Coordenades UTM del track
Dim puntTrack As coordUTM

'Datum en què està codificat el track
Dim datumTrack As New canviDatum

'Sobre l'arxiu
Dim arxiuTrack As New System.IO.StreamReader(dlgCarregarTrack.FileName)

Try
    'Es llegeix del fitxer, coneixent el seu format
    With arxiuTrack
        .ReadLine()
        nomDatumTrack = .ReadLine()

        'Es crea el canvi de datum que s'haurà de dur a terme, si escau
        If mdlVarsCompartides.datumActual = "WGS84" Then

            datumTrack.iniDatum("WGS84", "WGS84")

        Else

            datumTrack.iniDatum("WGS84", "ED50")

        End If

        'Línies sense informació útil
        .ReadLine()
        .ReadLine()
        .ReadLine()
        .ReadLine()

        'S'inicia el número de punts de track
        numPuntTrack = 0

        'Es llegeix la resta del fitxer, que conté totes les coordenades
        While arxiuTrack.EndOfStream = False

            'Es llegeix la línia actual i es separa per camps ","
            liniaActual = .ReadLine
            camps = liniaActual.Split(",")

            'El primer camp (latitud) i s'opera per a trobar el seu valor
            numèric
            coordenada = camps(0).Split(".")
            latitud = coordenada(0) + (coordenada(1) / Math.Pow(10,
            coordenada(1).Length))

            'El mateix per al segon camp (longitud)
            coordenada = camps(1).Split(".")
            longitud = coordenada(0) + (coordenada(1) / Math.Pow(10,
            coordenada(1).Length))

            'MsgBox(camps(0) & " <- camp0 radians i /10000000 camp1 -> " &
            camps(1))

            'MsgBox((camps(0) * (Math.PI / 180)) / 1000000 & " <- camp0
            radians i /10000000 camp1 -> " & (camps(1) * (Math.PI / 180)) / 1000000)

            'Es fa la transformació a coordenades UTM en radians
            puntTrack = geografiques2UTM(datumTrack, (latitud * (Math.PI /
            180)), (longitud * (Math.PI / 180)))
        End While
    End With
Catch

```

```

puntTrack.easting)      'MsgBox(puntTrack.northing & " <- northing    easting -> " &
                          'Per últim es transformen les coordenades UTM en coordenades
pantalla                UTM2coordPantalla(puntTrack, numPuntTrack)
                          numPuntTrack = numPuntTrack + 1

                          End While

                          End With

                          Catch ex As Exception
                              MsgBox(ex.ToString)

                          End Try

                          'S'actualitzen els valors del menú Track
                          mostrarOcultarTrack.Checked = True
                          mostrarTrack = True
                          redibuixarTrack()

                          End If

                          End Sub

Private Sub mostrarOcultarTrack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mostrarOcultarTrack.Click

    If mostrarTrack = False Then

        'El track no s'estava mostrant, per tant ara s'ha de mostrar
        mostrarTrack = True
        mostrarOcultarTrack.Checked = True

    Else

        'S'ha d'amagar el track
        mostrarTrack = False
        mostrarOcultarTrack.Checked = False

    End If

    'S'actualitza el mapa perquè dibuixi o esborri, segons el cas
    imgMapa.Refresh()
    pnlMapa_Scroll(Nothing, Nothing)

End Sub

'-----
'-----  FUNCIONS AUXILIARS  -----
'-----

Private Function geografiques2UTM(ByVal datum As canviDatum, ByVal latitud As
String, ByVal longitud As String) As coordUTM

    'Coordenades geodèsiques en radians
    Dim coordGeod As mdlVarsCompartides.coordenadaGeodesica

    coordGeod.latitud = latitud
    coordGeod.longitud = longitud

    'Es fa el canvi de coordenades corresponent
    If mdlVarsCompartides.datumActual = "WGS84" Then

        'Com el datum d'origen i destí és el mateix, no cal canvi de coordenades
        With datum

            Return .GeodesicaD2UTM(coordGeod)

        End With

    End If

```

```

Else

    'S'ha de realitzar el canvi de geodèsiques origen a UTM destí
    Return datum.GeodesicaO2UTMD(coordGeod)

End If

End Function

'Transformació de coordenades UTM a coordenades pantalla
Private Sub UTM2coordPantalla(ByVal puntUTM As coordUTM, ByVal dimensio As Double)

    If dimensio >= 0 Then

        'S'aplica la funció inversa a l'aplicada a l'hora de passar de coordenades
        'pantalla a coordenades UTM
        With costatsMapa

            arrayTrack(dimensio).Y = (Math.Abs(puntUTM.easting - .origenUTM.easting)
+ (.pasXDalt * (Math.Abs(puntUTM.northing - .origenUTM.northing)) / .pasYDalt)) / ((-
(.pasXDalt * .pasYEsquerra) / .pasYDalt) + .pasXEsquerra)
            arrayTrack(dimensio).X = (Math.Abs(puntUTM.easting - .origenUTM.easting)
+ (.pasXEsquerra * (Math.Abs(puntUTM.northing - .origenUTM.northing)) / .pasYEsquerra))
/ (.pasXDalt - ((.pasXEsquerra * .pasYDalt) / .pasYEsquerra))

            'MsgBox(arrayTrack(dimensio).X.ToString & " coord X - coord Y " &
arrayTrack(dimensio).Y.ToString)

        End With

    Else

        'Es calcula el punt actual que marca el GPS
        Dim temp As Point

        costatsMapaDatumCalibratge.calcularPasos(1)
        With costatsMapaDatumCalibratge

            'Es guarda en la variable temporal els valors obtinguts
            temp.X = Math.Floor((Math.Abs(puntUTM.easting - .origenUTM.easting) +
(.pasXEsquerra * (Math.Abs(puntUTM.northing - .origenUTM.northing)) / .pasYEsquerra)) /
(.pasXDalt - ((.pasXEsquerra * .pasYDalt) / .pasYEsquerra)))
            temp.Y = Math.Floor((Math.Abs(puntUTM.easting - .origenUTM.easting) +
(.pasXDalt * (Math.Abs(puntUTM.northing - .origenUTM.northing)) / .pasYDalt)) / ((-
(.pasXDalt * .pasYEsquerra) / .pasYDalt) + .pasXEsquerra))

            'Es comparen amb els que tenim guardats i si hi ha canvi en algun valor
            'es refresca el mapa
            If temp.X <> coordPantallaGPS.X OrElse temp.Y <> coordPantallaGPS.Y Then

                imgMapa.Refresh()

            End If

            'S'assignen els nous valors
            coordPantallaGPS.X = temp.X
            coordPantallaGPS.Y = temp.Y

        End With

    End If

End Sub

#End Region

#Region "Menú Eines"

    Private Sub einesCanviCoord_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles einesCanviCoord.Click

        If canviCoordenades = False Then

            canviCoordenades = True


```

```

        Dim frmCanvi As New frmCanviCoord

        frmCanvi.Show()

    End If

End Sub

#End Region

#Region "Tractament del mapa"

    Private Sub imgMapa_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles imgMapa.MouseDown

        If imgCarregada = True Then

            posicioClickInicial = New Point(e.X, e.Y)

        End If

    End Sub

    Private Sub imgMapa_MouseMove(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles imgMapa.MouseMove

        'Es captura la posició del mouse en tot moment
        puntImatge = imgMapa.PointToClient(Windows.Forms.Cursor.Position)

        'Es comprova si es vol desplaçar el mapa amb el mouse (botó esquerre)
        If e.Button = Windows.Forms.MouseButtons.Left Then

            'L'autoscroll torna números negatius, per això es multiplica * -1
            Dim DeltaX As Integer = (e.X - posicioClickInicial.X) * -1
            Dim DeltaY As Integer = (e.Y - posicioClickInicial.Y) * -1

            pnlMapa.AutoScrollPosition = New Drawing.Point((-1 *
pnlMapa.AutoScrollPosition.X) + DeltaX, (-1 * pnlMapa.AutoScrollPosition.Y) + DeltaY)

        End If

    End Sub

    Private Sub imgMapa_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles imgMapa.Click

        'En cas que estigui calibrant-se el mapa
        If puntsCalibratge = True Then

            'Es captura la posició actual del mouse
            puntUltimClick = imgMapa.PointToClient(Windows.Forms.Cursor.Position)

            'S'aplica el factor d'escala aplicat
            puntUltimClick.X /= factorEscala
            puntUltimClick.Y /= factorEscala

            'S'assigna aquest punt a la pestanya activa de calibratge
            arrayPuntsCalibratge(pestanyaActiva).coordPantalla = puntUltimClick

            'S'actualitza el mapa per reflectir els canvis
            imgMapa.Refresh()

        End If

        If mostrarTrack = True Then

            'Es redibuixa el track cada cop que es clica al mapa, per si de cas
            redibuixarTrack()

        End If

    End Sub

```

```

Private Sub pnlMapa_Scroll(ByVal sender As Object, ByVal e As
System.Windows.Forms.ScrollEventArgs) Handles pnlMapa.Scroll

    If mostrarTrack = True Then

        'Es redibuixa el track perquè apareguin les parts ocultes possibles
        redibuixarTrack()

    End If

End Sub

'-----
'----- FUNCIONS AUXILIARS -----
'-----

Sub redibuixarTrack()

    'S'actualitza el mapa
    Refresh()

    Dim pinzell As New Pen(Color.Red, 2)
    Dim centrePinzell As New Point
    Dim puntActual As New Point
    Dim puntSeguent As New Point

    'Es dibuixa cada punt del track tenint en compte el factor d'escala
    For i As Integer = 0 To numPuntTrack - 2 Step 1

        puntActual.X = arrayTrack(i).X * factorEscala
        puntActual.Y = arrayTrack(i).Y * factorEscala

        puntSeguent.X = arrayTrack(i + 1).X * factorEscala
        puntSeguent.Y = arrayTrack(i + 1).Y * factorEscala

        imgMapa.CreateGraphics.DrawLine(New Pen(Color.Red, 4), puntActual,
puntSeguent)

    Next

End Sub

Private Sub redibuixarPunts()

    Dim gruixPinzell As Integer = 9

    'Colors dels punts de calibratge
    Dim colorsPunts() As Color = {Color.Red, Color.Blue, Color.Green}

    'Per cada punt de calibratge
    For i As Integer = 0 To numPuntsCalibratge - 1 Step 1

        With arrayPuntsCalibratge(i)

            Try

                'En cas que ja s'hagi clicat en algun punt del mapa
                If .coordPantalla.X <> 0 AndAlso .coordPantalla.Y <> 0 Then

                    Dim pinzell As New Pen(colorsPunts(i), (gruixPinzell / 3))
                    Dim centrePinzell As New Point

                    centrePinzell.X = Math.Floor((.coordPantalla.X * factorEscala) -
(gruixPinzell / 2))
                    centrePinzell.Y = Math.Floor((.coordPantalla.Y * factorEscala) -
(gruixPinzell / 2))

                    imgMapa.CreateGraphics.DrawEllipse(pinzell, New
Rectangle(centrePinzell, New Size(gruixPinzell, gruixPinzell))

                    centrePinzell.X = Math.Floor((.coordPantalla.X * factorEscala) -
gruixPinzell)
                    centrePinzell.Y = Math.Floor((.coordPantalla.Y * factorEscala) -
gruixPinzell)

                End Try

            End With

        End For

    End Sub

```

```

        imgMapa.CreateGraphics.DrawEllipse(pinzell, New
Rectangle(centrePinzell, New Size(gruixPinzell * 2, gruixPinzell * 2)))

        End If

        Catch ex As Exception

        End Try

    End With

Next

Try

    'Si està iniciada la comunicació, es dibuixa la marca d'ubicació actual
    If iniciarCOM.Enabled = False Then

        Dim pinzell As New Pen(Color.Purple, 3)
        Dim centrePinzell As New Point

        'Es centra la pantalla per veure la posició actual
        centrarPantalla()

        'Es dibuixen les marques
        centrePinzell.X = Math.Floor((coordPantallaGPS.X * factorEscala) -
(gruixPinzell * 2))
        centrePinzell.Y = Math.Floor((coordPantallaGPS.Y * factorEscala) -
(gruixPinzell * 2))

        imgMapa.CreateGraphics.DrawEllipse(pinzell, New Rectangle(centrePinzell,
New Size(gruixPinzell * 4, gruixPinzell * 4)))

        centrePinzell.X = Math.Floor((coordPantallaGPS.X * factorEscala) -
(gruixPinzell))
        centrePinzell.Y = Math.Floor((coordPantallaGPS.Y * factorEscala) -
(gruixPinzell))

        imgMapa.CreateGraphics.DrawEllipse(pinzell, New Rectangle(centrePinzell,
New Size(gruixPinzell * 2, gruixPinzell * 2)))

        'Punts que defineixen el polígon de la fletxa
        Dim fletxa(7) As PointF

        fletxa(0).X = (10 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(0).Y = 0 + coordPantallaGPS.Y * factorEscala
        fletxa(1).X = (0 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(1).Y = 10 + coordPantallaGPS.Y * factorEscala
        fletxa(2).X = (7.5 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(2).Y = 10 + coordPantallaGPS.Y * factorEscala
        fletxa(3).X = (7.5 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(3).Y = 25 + coordPantallaGPS.Y * factorEscala
        fletxa(4).X = (12 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(4).Y = 25 + coordPantallaGPS.Y * factorEscala
        fletxa(5).X = (12 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(5).Y = 10 + coordPantallaGPS.Y * factorEscala
        fletxa(6).X = (20 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(6).Y = 10 + coordPantallaGPS.Y * factorEscala
        fletxa(7).X = (10 + coordPantallaGPS.X - 10) * factorEscala
        fletxa(7).Y = 0 + coordPantallaGPS.Y * factorEscala

        imgMapa.CreateGraphics.FillPolygon(Brushes.Red, fletxa)

    End If

    Catch ex As Exception

    End Try

End Sub

Sub centrarPantalla()

    'L'autoscroll torna números negatius, per això es multiplica * -1

```

```

        Dim DeltaX As Integer = (coordPantallaGPS.X * factorEscala -
        (pnlMapa.ClientSize.Width / 2)) * -1
        Dim DeltaY As Integer = (coordPantallaGPS.Y * factorEscala -
        (pnlMapa.ClientSize.Height / 2)) * -1

        pnlMapa.AutoScrollPosition = New Drawing.Point((-1 * DeltaX), (-1 * DeltaY))

    End Sub

#End Region

#Region "Timers"

    Public Sub TimerPuntsDisp(ByVal sender As Object, ByVal e As
    System.Timers.ElapsedEventArgs)

        redibuixarPunts()

    End Sub

    Public Sub TimerInfoDisp(ByVal sender As Object, ByVal e As
    System.Timers.ElapsedEventArgs)

        'Si hi ha un mapa carregat
        If imgCarregada = True Then

            'S'escriu el valor de la posició actual del mouse
            coordX.Text = Math.Floor(puntImatge.X.ToString / factorEscala)
            coordY.Text = Math.Floor(puntImatge.Y.ToString / factorEscala)

            'S'actualitzen els valors de les propietats de calibratge
            projeccio.Text = projeccioCalibratge
            datum.Text = datumCalibratge
            punts.Text = numPuntsCalibratge

            'Quan s'ha acceptat un calibratge
            If infoUTM = True Then

                'S'inicia el timer de les dades UTM
                infoUTM = False
                timerUTM.Start()
                costatsMapa.calcularPasos(0)

            End If

        Else

            'En cas que no hagi mapa carregat s'aturen els timers
            timerInfo.Stop()
            timerUTM.Stop()
            timerPunts.Stop()

        End If

    End Sub

    Public Sub TimerUTMDisp(ByVal sender As Object, ByVal e As
    System.Timers.ElapsedEventArgs)

        calcularPosicioUTM()

    End Sub

'-----
'----- FUNCIONS AUXILIARS -----
'-----

    Public Sub calcularPosicioUTM()

        'S'apliquen les fórmules obtingudes per al càlcul de la posició
        'actual del mouse en coordenades UTM
        With costatsMapa.origenUTM

```

```

        zona.Text = .zona
        hemisferi.Text = .hemisferi

        easting.Text = Math.Floor(.easting + costatsMapa.pasXDalt * (puntImatge.X /
factorEscala) + costatsMapa.pasXEsquerra * (puntImatge.Y / factorEscala))
        northing.Text = Math.Floor(.northing + costatsMapa.pasYESquerra *
(puntImatge.Y / factorEscala) + costatsMapa.pasYDalt * (puntImatge.X / factorEscala))

    End With

End Sub

#End Region

#Region "Formulari principal"

Private Sub frmPrincipal_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    'S'inicien els timers
    AddHandler timerPunts.Elapsed, AddressOf TimerPuntsDisp
    AddHandler timerInfo.Elapsed, AddressOf TimerInfoDisp
    AddHandler timerUTM.Elapsed, AddressOf TimerUTMDisp

    'S'inicia un port sèrie per defecte
    With pSerie

        .BaudRate = 9600
        .Parity = 0
        '.PortName = System.IO.Ports.SerialPort.GetPortNames(0)
        .PortName = "COM8"
        .StopBits = 1
        .ReadTimeout = 1500

    End With

    'frmGPRMC.Show()

    'Es manté la compatibilitat amb VB.NET 2003
    CheckForIllegalCrossThreadCalls = False

End Sub

Private Sub frmPrincipal_LocationChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.LocationChanged

    If mostrarTrack = True Then

        redibuixarTrack()

    End If

End Sub

#End Region

End Class

```

1.2. frmPropietatsCalibratge.vb

```

Public Class frmPropietatsCalibratge
    Inherits System.Windows.Forms.Form

    Private Sub frmPropietatsCalibratge_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Calibratge = True
    End Sub

```

```

Private Sub btnAcceptar_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAcceptar.Click

    If Me.datum.SelectedItem Is "" OrElse Me.projeccio.SelectedItem Is "" Then
        MsgBox("Si us plau, tria un datum i una projecció vàlids",
MsgBoxStyle.Exclamation, "Dades incompletes")
    Else
        If puntsCalibratge = False Then
            numPuntsCalibratge = Me.numeroPunts.Value
            datumCalibratge = Me.datum.SelectedItem
            projeccioCalibratge = Me.projeccio.SelectedItem

            Dim frmPunts As New frmPuntsCalibratge

            Calibratge = False
            jaCalibrat = False

            frmPunts.Show()

            Close()
            Dispose()

        End If
    End If

End Sub

Private Sub btnCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancelar.Click
    Calibratge = False
    Dispose()
End Sub

Private Sub frmPropietatsCalibratge_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    Calibratge = False
    Dispose()
End Sub

End Class

```

1.3. frmPuntsCalibratge.vb

```

Public Class frmPuntsCalibratge
    Inherits System.Windows.Forms.Form

    Private timer As New System.Timers.Timer(15)
    Private arrayPuntsCalibratgeTemporal() As puntCalibratge

    #Region "Events de càrrega i tancament del formulari"

    Private Sub frmPuntsCalibratge_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Notifiquem que és visible aquesta finestra
        puntsCalibratge = True
        frmPrincipal.datumActual.Text = datumCalibratge

        'La posicionem
        Me.Left = Screen.PrimaryScreen.WorkingArea.Width - 250
        Me.Top = 150

        'Si ja ha estat calibrat aquest mapa, carreguem les dades
        If jaCalibrat = True Then

            recalibrar()

```

```

Else

    'Sinó definim la mida de l'array de punts
    ReDim arrayPuntsCalibratge(numPuntsCalibratge - 1)
    ReDim arrayPuntsDatumActual(numPuntsCalibratge - 1)

End If

'Guardem les dades actuals, per si es cancel·la
ReDim arrayPuntsCalibratgeTemporal(numPuntsCalibratge - 1)

For i As Integer = 0 To numPuntsCalibratge - 1

    arrayPuntsCalibratgeTemporal(i) = arrayPuntsCalibratge(i)

Next

'Fem que la pestanya del punt 1 sigui la que es mostri
TabControl1.SelectedTab = Me.TabPage1
pestanyaActiva = 0

'Ajustem el timer i l'iniciem
AddHandler timer.Elapsed, AddressOf TimerFired
timer.Start()

'Refresquem la imatge per esborrar possibles marques anteriors
frmPrincipal.imgMapa.Refresh()

End Sub

#End Region

#Region "Timer"

Private Sub TimerFired(ByVal sender As Object, ByVal e As
System.Timers.ElapsedEventArgs)

    If imgCarregada = False Then

        Me.Close()
        Exit Sub

    End If

    pestanyaActiva = Me.TabControl1.SelectedIndex

    Select Case pestanyaActiva
        Case 0
            coordx1.Text = arrayPuntsCalibratge(0).coordPantalla.X.ToString
            coordy1.Text = arrayPuntsCalibratge(0).coordPantalla.Y.ToString
        Case 1
            coordx2.Text = arrayPuntsCalibratge(1).coordPantalla.X.ToString
            coordy2.Text = arrayPuntsCalibratge(1).coordPantalla.Y.ToString
        Case 2
            coordx3.Text = arrayPuntsCalibratge(2).coordPantalla.X.ToString
            coordy3.Text = arrayPuntsCalibratge(2).coordPantalla.Y.ToString

    End Select

End Sub

#End Region

#Region "Tab 1"

Private Sub btnSeguent1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSeguent1.Click

    TabControl1.SelectedTab = Me.TabPage2

End Sub

```

```

#End Region

#Region "Tab 2"

    Private Sub btnAnterior2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAnterior2.Click

        TabControl1.SelectedTab = Me.TabPage1

    End Sub

    Private Sub btnSeguent2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSeguent2.Click

        TabControl1.SelectedTab = Me.TabPage3

    End Sub

#End Region

#Region "Tab 3"

    Private Sub btnAnterior3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAnterior3.Click

        TabControl1.SelectedTab = Me.TabPage2

    End Sub

#End Region

#Region "Botons"

    'Botó: "Guardar Calibratge"
    Private Sub btnGuardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGuardar.Click

        guardarCalibratge()

    End Sub

    'Botó: "Acceptar"
    Private Sub btnAcceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAcceptar.Click

        acceptarCalibratge()

        timer.Stop()

        'Actualitzem noves opcions del menú principal
        frmPrincipal.menuCOM.Enabled = True
        frmPrincipal.mostrarOcultarTrack.Checked = False

        'Notifiquem que tanquem la finestra
        puntsCalibratge = False
        Close()

    End Sub

    'Botó: "Descartar"
    Private Sub btnCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancelar.Click

        'Aturem el timer
        timer.Stop()

        puntUltimClick = Point.Empty

        'Restauem el valor anterior dels punts, ja que no volem modificar
        arrayPuntsCalibratge = arrayPuntsCalibratgeTemporal
        frmPrincipal.imgMapa.Refresh()

        'Notifiquem que no serà visible aquesta finestra

```

```

        puntsCalibratge = False
        Close()

    End Sub

#End Region

#Region "Sub auxiliars"

'Salvem les dades que hi ha introduïdes als formularis de manera temporal
Private Sub acceptarCalibratge()

    Try

        arrayPuntsCalibratge(0).easting = easting1.Text
        arrayPuntsCalibratge(0).northing = northing1.Text
        arrayPuntsCalibratge(0).hemisferi = ns1.Text
        arrayPuntsCalibratge(0).zona = zona1.Text

        arrayPuntsCalibratge(1).easting = easting2.Text
        arrayPuntsCalibratge(1).northing = northing2.Text
        arrayPuntsCalibratge(1).hemisferi = ns2.Text
        arrayPuntsCalibratge(1).zona = zona2.Text

        arrayPuntsCalibratge(2).easting = easting3.Text
        arrayPuntsCalibratge(2).northing = northing3.Text
        arrayPuntsCalibratge(2).hemisferi = ns3.Text
        arrayPuntsCalibratge(2).zona = zona3.Text

        'Treballem amb l'array de punts que es triï en cada moment
        For i As Integer = 0 To numPuntsCalibratge - 1

            arrayPuntsDatumActual(i) = arrayPuntsCalibratge(i)

        Next

    Catch ex As Exception

        'Si entra aquí no podem donar com a bo el calibratge
        MsgBox(ex.Message & Chr(13) & "Si us plau, comprova la integritat de les
dades.", _
            MsgBoxStyle.Critical, "Excepció detectada")

        jaCalibrat = False
        infoUTM = False
        Exit Sub

    End Try

    'Si arriba a aquest punt, les dades eren correctes
    jaCalibrat = True
    infoUTM = True

    frmPrincipal.menuDatum.Enabled = True
    frmPrincipal.datumActual.Text = datumCalibratge

End Sub

'Procediment que guarda les dades en un fitxer amb nom que introdueix l'usuari
Private Sub guardarCalibratge()

    'Comprovem que el mapa ha estat ben calibrat
    'Amb coordenades diferents de (0,0) al mapa
    For j As Integer = 0 To numPuntsCalibratge - 1 Step 1

        With arrayPuntsCalibratge(j)

            If .coordPantalla.X = 0 AndAlso .coordPantalla.Y = 0 Then

                MsgBox("Si us plau, introdueix tots els punts abans de guardar",
MsgBoxStyle.Exclamation, "Dades de punts invàlides")
                Exit Sub

            End If

        End With

    Next j

End Sub

```

```

        End With

    Next

    'Comprovem que totes les dades UTM han estat introduïdes
    If zona1.Text = "" OrElse zona2.Text = "" OrElse zona3.Text = "" OrElse _
        northing1.Text = "" OrElse northing2.Text = "" OrElse northing3.Text = ""
    OrElse _
        easting1.Text = "" OrElse easting2.Text = "" OrElse easting3.Text = "" Then

        MsgBox("Si us plau, introdueix totes les dades UTM abans de guardar",
        MsgBoxStyle.Exclamation, "Dades UTM invàlides")
        Exit Sub

    End If

    Dim dlgGuardarPunts As New SaveFileDialog

    dlgGuardarPunts.Filter = "Arxiu de calibratge (*.dat) | *.dat"

    If dlgGuardarPunts.ShowDialog() = Windows.Forms.DialogResult.OK Then

        Try

            Dim arxiu As New System.IO.StreamWriter(dlgGuardarPunts.FileName)

            With arxiu
                .WriteLine(datumCalibratge)
                .WriteLine(projeccioCalibratge)
                .WriteLine(numPuntsCalibratge)

                For i As Integer = 0 To numPuntsCalibratge - 1 Step 1

                    .WriteLine(arrayPuntsCalibratge(i).coordPantalla.X)
                    .WriteLine(arrayPuntsCalibratge(i).coordPantalla.Y)

                Next

                .WriteLine(zona1.Text)
                .WriteLine(easting1.Text)
                .WriteLine(northing1.Text)
                .WriteLine(ns1.Text)

                .WriteLine(zona2.Text)
                .WriteLine(easting2.Text)
                .WriteLine(northing2.Text)
                .WriteLine(ns2.Text)

                .WriteLine(zona3.Text)
                .WriteLine(easting3.Text)
                .WriteLine(northing3.Text)
                .WriteLine(ns3.Text)

                .Close()

            End With

            Catch ex As Exception

                MsgBox(ex.ToString)
                Exit Sub

            End Try

            MsgBox("El fitxer de calibratge ha estat guardat satisfactòriament.",
            MsgBoxStyle.Information, "Fitxer Guardat")

        Else

            MsgBox("El fitxer de calibratge no ha estat guardat.",
            MsgBoxStyle.Exclamation, "Atenció!")

        End If

    End Sub

```

```
'Recuperem els valors guardats prèviament i els mostrem al formulari
Private Sub recalibrar()

    Try
        easting1.Text = arrayPuntsCalibratge(0).easting.ToString
        northing1.Text = arrayPuntsCalibratge(0).northing.ToString
        ns1.Text = arrayPuntsCalibratge(0).hemisferi.ToString
        zona1.Text = arrayPuntsCalibratge(0).zona.ToString

        easting2.Text = arrayPuntsCalibratge(1).easting.ToString
        northing2.Text = arrayPuntsCalibratge(1).northing.ToString
        ns2.Text = arrayPuntsCalibratge(1).hemisferi.ToString
        zona2.Text = arrayPuntsCalibratge(1).zona.ToString

        easting3.Text = arrayPuntsCalibratge(2).easting.ToString
        northing3.Text = arrayPuntsCalibratge(2).northing.ToString
        ns3.Text = arrayPuntsCalibratge(2).hemisferi.ToString
        zona3.Text = arrayPuntsCalibratge(2).zona.ToString

    Catch ex As Exception

        MsgBox(ex.Message & Chr(13) & "Si us plau, comprova la integritat de les
dades.", _
            MsgBoxStyle.Critical, "Excepció detectada")

    End Try

End Sub

#End Region

End Class
```

1.4. frmConfigCom.vb

```
Imports PFC.mdlVarsCompartides

Public Class frmConfigCom
    Inherits System.Windows.Forms.Form

    Private Sub frmConfigCom_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        configCOM = True

        With frmPrincipal.pSerie

            For i As Integer = 0 To _
My.Computer.Ports.SerialPortNames.Count - 1
                portCOM.Items.Add( _
                    My.Computer.Ports.SerialPortNames(i))
            Next
            portCOM.Text = .PortName

            Select Case .Parity
                Case 0
                    paritat.Text = "Cap"
                Case 1
                    paritat.Text = "Senar"
                Case 2
                    paritat.Text = "Parell"
            End Select

            bitsStop.Text = .StopBits
            bauds.Text = .BaudRate

        End With
```

```

End Sub

Private Sub btnTest_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTest.Click

    Try
        If frmPrincipal.pSerie.IsOpen Then

            MessageBox.Show("Port " & portCOM.Text & " NO disponible.", "Test",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        Else

            MessageBox.Show("Port " & portCOM.Text & " disponible.", "Test",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If

    Catch ex As Exception

        MessageBox.Show("Prova fallada!", "Test", MessageBoxButtons.OK,
MessageBoxIcon.Error)

    End Try

End Sub

Private Sub btnAceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAceptar.Click

    With frmPrincipal.pSerie

        If .IsOpen Then

            .Close()

        End If

        .PortName = portCOM.Text

        Select Case paritat.SelectedIndex
            Case 0
                .Parity = 0
            Case 1
                .Parity = 2
            Case 2
                .Parity = 1
        End Select

        .StopBits = bitsStop.Text
        .BaudRate = bauds.Text

    End With

    configCOM = False
    Dispose()

End Sub

Private Sub btnCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancelar.Click

    configCOM = False
    Dispose()

End Sub

End Class

```

1.5. frmCanviCoord.vb

```
Imports PFC.canvidatum

Public Class frmCanviCoord
    Inherits System.Windows.Forms.Form

    Private Sub btnTransformar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTransformar.Click

        If Me.cEasting.Text = "" OrElse Me.cNorthing.Text = "" OrElse Me.cZona.Text = ""
Then
            MsgBox("Si us plau, emplena tots els camps.", MsgBoxStyle.Information)

        ElseIf Me.dED50.Checked = False Then
            MsgBox("Si us plau, selecciona un datum de destí.", MsgBoxStyle.Information)
        Else
            Try
                cUTM.easting = Me.cEasting.Text
                cUTM.northing = Me.cNorthing.Text
                cUTM.zona = Me.cZona.Text
                cUTM.hemisferi = Me.cHemisferi.Text
            Catch ex As Exception
                MsgBox(ex.Message & Chr(13) & "Si us plau, comprova la integritat de les
dades.", _
                    MsgBoxStyle.Critical, "Excepció detectada")
            End Try

            Dim datum As New canviDatum
            Dim cUTMD As New coordUTM

            If dED50.Checked Then
                datum.iniDatum("WGS84", "ED50")

                cUTMD = datum.canviDeDatum(cUTM)

                MsgBox("Easting: " & cUTMD.easting & Chr(13) _
                    & "Northing: " & cUTMD.northing & Chr(13) _
                    & "Zona UTM: " & cUTMD.zona & Chr(13) _
                    & "Hemisferi: " & cUTMD.hemisferi, _
                    MsgBoxStyle.Information, "Resultat")
            End If

        End If

    End Sub

    Private Sub btnTancar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTancar.Click

        Dispose()
        canviCoordenades = False

    End Sub

End Class
```

2. CLASSES

2.1. elipsoide.vb

```

Friend Class elipsoide
    Public a As Double      'Semieix major
    Public b As Double      'Semieix menor
    Public excentricitat As Double
    Public segonaExcentricitat As Double
    Public segonaExcQuadrat As Double
    Public c As Double      'Radi polar de corbatura
    Public alpha As Double  'Aixafament de l'el·lipsoide
    Public h As Double      'Altitud el·lipsoïdal. Considerarem sempre 771.76 en WGS84
    Public N As Double      'Radi de corbatura a la vertical principal del punt

    'Iniciem un el·lipsoide amb les seves dades
    Public Function inicialitzarElipsoide(ByVal datum As String) As elipsoide

        Dim elip As New elipsoide

        Select Case datum
            'Iniciem dades de l'el·lipsoide de Hayford 1924"
            Case "ED50"

                With elip
                    .a = 6378388
                    .b = 6356911.94613
                End With

                'Dades de l'el·lipsoide del WGS84
            Case "WGS84"

                With elip
                    .a = 6378137
                    .b = 6356752.314245
                    .h = 771.76
                End With

            End Select

            With elip
                .excentricitat = (Math.Sqrt((Math.Pow(.a, 2) - Math.Pow(.b, 2))) / .a)
                .segonaExcentricitat = (Math.Sqrt((Math.Pow(.a, 2) - Math.Pow(.b, 2))) / .b)
                .segonaExcQuadrat = Math.Pow(.segonaExcentricitat, 2)
                .c = Math.Pow(.a, 2) / .b
                .alpha = (.a - .b) / .a
            End With

            Return elip

        End Function

        Public Sub New()

        End Sub

    End Class

```

2.2. canvidatum.vb

```

Friend Class canviDatum

    'El·lipsoide associat al datum d'origen
    Public nomOrigen As String

```

```

Public elipsOrigen As New elipsoide

'El·lipsoide associat al datum de destí
Public nomDesti As String
Public elipsDesti As New elipsoide

'Paràmetres de transformació de datums
Public AX As Double      'Termes de translació
Public AY As Double
Public AZ As Double
Public RX As Double      'Termes de rotació
Public RY As Double
Public RZ As Double
Public e As Double       'Factor d'escala

Public Sub New()
End Sub

'Inicialitza datum amb les dades de l'el·lipsoide
Public Sub iniDatum(ByVal datumOrigen As String, ByVal datumDesti As String)

    nomOrigen = datumOrigen
    elipsOrigen = elipsOrigen.inicialitzarElipsoide(datumOrigen)

    nomDesti = datumDesti
    elipsDesti = elipsDesti.inicialitzarElipsoide(datumDesti)

End Sub

Public Function canviDeDatum(ByVal coordUTMOrigen As coordUTM) As coordUTM

    Dim coordGeodO As New coordenadaGeodesica
    Dim coordGeocO As New coordenadaGeocentrica
    Dim coordGeocD As New coordenadaGeocentrica
    Dim coordGeodD As New coordenadaGeodesica

    coordGeodO = UTM2GeodesicaO(coordUTMOrigen)
    coordGeocO = GeodesiquesO2GeocentricesO(coordGeodO)
    coordGeocD = GeocentricaO2GeoCentricaD(coordGeocO)
    coordGeodD = GeocentricaD2GeodesicaD(coordGeocD)

    Return GeodesicaD2UTM(coordGeodD)

End Function

Public Function GeodesicaO2UTMD(ByVal coordGeodOrigen As coordenadaGeodesica) As
coordUTM

    Dim coordGeocO As New coordenadaGeocentrica
    Dim coordGeocD As New coordenadaGeocentrica
    Dim coordGeodD As New coordenadaGeodesica

    coordGeocO = GeodesiquesO2GeocentricesO(coordGeodOrigen)
    coordGeocD = GeocentricaO2GeoCentricaD(coordGeocO)
    coordGeodD = GeocentricaD2GeodesicaD(coordGeocD)

    Return GeodesicaD2UTM(coordGeodD)

End Function

Private Function UTM2GeodesicaO(ByVal UTM As coordUTM) As coordenadaGeodesica

    '-----
    'Pas UTMorigen -> GeodèsiquesOrigen
    '-----
    Dim coordGeod As New coordenadaGeodesica

    Dim lambda As Double
    Dim phi As Double

```

```

Dim v As Double
Dim a0 As Double
Dim A1 As Double
Dim A2 As Double
Dim J2 As Double
Dim J4 As Double
Dim J6 As Double
Dim alfa As Double
Dim beta As Double
Dim gamma As Double
Dim Bphi As Double
Dim b0 As Double
Dim zeta As Double
Dim xi As Double
Dim eta As Double
Dim Alambda As Double
Dim tau As Double
Dim cTemp As Double

'Tractament previ a les coordenades UTM
UTM.easting = UTM.easting - 500000

If UTM.hemisferi = "S" Then

    UTM.northing = UTM.northing - 10000000

End If

lambda = (UTM.zona * 6) - 183

'Càlcul dels paràmetres de UTM a Geodèsiques (lat, long, h)
With elipsOrigen

    phi = UTM.northing / (6366197.724 * 0.9996)
    v = (.c / Math.Sqrt(1 + .segonaExcQuadrat * cosQuadrat(phi))) * 0.9996
    a0 = UTM.easting / v
    A1 = Math.Sin((2 * phi))
    A2 = A1 * cosQuadrat(phi)
    J2 = phi + (A1 / 2)
    J4 = (3 * J2 + A2) / 4
    J6 = (5 * J4 + A2 * cosQuadrat(phi)) / 3
    alfa = (3 / 4) * .segonaExcQuadrat
    beta = (5 / 3) * Math.Pow(alfa, 2)
    gamma = (35 / 27) * Math.Pow(alfa, 3)
    Bphi = 0.9996 * .c * (phi - alfa * J2 + beta * J4 - gamma * J6)
    b0 = (UTM.northing - Bphi) / v
    zeta = ((.segonaExcQuadrat * Math.Pow(a0, 2)) / 2) * cosQuadrat(phi)
    xi = a0 * (1 - (zeta / 3))
    eta = b0 * (1 - zeta) + phi
    Alambda = Math.Atan((Math.Sinh(xi) / Math.Cos(eta)))
    tau = Math.Atan(Math.Cos(Alambda) * Math.Tan(eta))

    'Càlcul de la longitud
    cTemp = ((Alambda * 180) / Math.PI + lambda) * (Math.PI / 180)
    coordGeod.longitud = cTemp

    'Càlcul de la latitud
    cTemp = ((phi) + (1 + .segonaExcQuadrat * cosQuadrat(phi) - _
        (3 / 2) * .segonaExcQuadrat * Math.Sin(phi) * Math.Cos(phi) - _
        * (tau - phi)) * (tau - phi))
    coordGeod.latitud = cTemp

End With

Return coordGeod

End Function

Private Function Geodesiques02Geocentriques0(ByVal coordgeod As coordenadaGeodesica)
As coordenadaGeocentrica

    '-----
    'Pas de Geodèsiques Origen -> Geocèntriques Origen
    '-----
    Dim coordGeoc As New coordenadaGeocentrica

```

```

With elipsOrigen

    .N = Math.Pow(.a, 2) / (Math.Sqrt((Math.Pow(.a, 2) *
cosQuadrat(coordgeod.latitud) _
    + (Math.Pow(.b, 2) * sinQuadrat(coordgeod.latitud))))))

    coordGeoc.X = (.N + .h) * Math.Cos(coordgeod.latitud) *
Math.Cos(coordgeod.longitud)
    coordGeoc.Y = (.N + .h) * Math.Cos(coordgeod.latitud) *
Math.Sin(coordgeod.longitud)
    coordGeoc.Z = ((Math.Pow(.b, 2) / Math.Pow(.a, 2)) * .N + .h) *
Math.Sin(coordgeod.latitud)

End With

Return coordGeoc

End Function

Private Function GeocentricaO2GeoCentricaD(ByVal coordGeocO As
coordenadaGeocentrica) As coordenadaGeocentrica

    iniciarParametresTransformacio()

    '-----
    'Pas de Geocèntriques datum original -> Geocèntriques nou datum
    '-----
    Dim coordGeocD As New coordenadaGeocentrica

    With coordGeocO

        coordGeocD.X = AX + (1 + e) * (1 * .X + RZ * .Y - RY * .Z)
        coordGeocD.Y = AY + (1 + e) * (-RZ * .X + 1 * .Y + RX * .Z)
        coordGeocD.Z = AZ + (1 + e) * (RY * .X - RX * .Y + 1 * .Z)

    End With

    Return coordGeocD

End Function

'Inicialitza els paràmetres del mètode Bursa-Wolf
Private Sub iniciarParametresTransformacio()

    'De WGS84 a ED50
    If nomOrigen = "WGS84" AndAlso nomDesti = "ED50" Then

        AX = 136.666957
        AY = 145.34122
        AZ = 173.15534

        RX = -0.000010523589
        RY = -0.000000903055
        RZ = -0.000000048715
        e = -0.000011938705

    End If

    'De ED50 a WGS84
    If nomOrigen = "ED50" AndAlso nomDesti = "WGS84" Then

        AX = -136.666957
        AY = -145.34477
        AZ = -173.156

        RX = 0.000010523589
        RY = 0.000000903055
        RZ = 0.000000048715
        e = 0.000011938847

    End If

End Sub

```

```

Private Function GeocentricaD2GeodesicaD(ByVal coordGeocD As coordenadaGeocentrica)
As coordenadaGeodesica

    '''-----
    '''Pas de Geocèntriques datum destí -> Geodèsiques datum destí
    '''-----

    Dim coordGeodD As New coordenadaGeodesica
    Dim theta As New Double 'Factor d'escala
    Dim p As New Double 'Segon factor d'escala

    With coordGeocD

        p = Math.Sqrt(Math.Pow(.X, 2) + Math.Pow(.Y, 2))
        theta = Math.Atan((.Z * elipsDesti.a) / (p * elipsDesti.b))
        coordGeodD.longitud = Math.Atan(.Y / .X)

    End With

    With elipsDesti

        coordGeodD.latitud = Math.Atan((coordGeocD.Z + (.segonaExcQuadrat * .b *
(sinQuadrat(theta) * Math.Sin(theta)))) / (p - (Math.Pow(.excentricitat, 2) * .a * (cosQuadrat(theta) *
Math.Cos(theta)))))

        .N = Math.Pow(.a, 2) / (Math.Sqrt((Math.Pow(.a, 2) *
cosQuadrat(coordGeodD.latitud) + (Math.Pow(.b, 2) * sinQuadrat(coordGeodD.latitud)))))

        .h = (p / Math.Cos(coordGeodD.latitud)) - .N

    End With

    Return coordGeodD

End Function

Friend Function GeodesicaD2UTM(ByVal coordGeodD As coordenadaGeodesica) As coordUTM

    Dim pUTM As New coordUTM
    Dim longDec As Double = coordGeodD.longitud * (180 / Math.PI)
    Dim latDec As Double = coordGeodD.latitud * (180 / Math.PI)

    Dim lambda0 As Double
    Dim Alambda As Double
    Dim A As Double
    Dim xi As Double
    Dim eta As Double
    Dim v As Double
    Dim zeta As Double
    Dim A1 As Double
    Dim A2 As Double
    Dim J2 As Double
    Dim J4 As Double
    Dim J6 As Double
    Dim alfa As Double
    Dim beta As Double
    Dim gamma As Double
    Dim Bphi As Double

    pUTM.zona = Math.Floor((longDec / 6) + 31)
    lambda0 = (pUTM.zona * 6 - 183) * (Math.PI / 180)

    With coordGeodD

        Alambda = .longitud - lambda0
        A = Math.Cos(.latitud) * Math.Sin(Alambda)
        xi = (1 / 2) * Math.Log((1 + A) / (1 - A))
        eta = Math.Atan(Math.Tan(.latitud) / Math.Cos(Alambda)) - .latitud
        v = (elipsDesti.c / Math.Sqrt(1 + elipsDesti.segonaExcQuadrat *
cosQuadrat(.latitud))) * 0.9996
        zeta = (elipsDesti.segonaExcQuadrat / 2) * Math.Pow(xi, 2) *
cosQuadrat(.latitud)
        A1 = Math.Sin((2 * .latitud))
    End With

```

```

        A2 = A1 * cosQuadrat(.latitud)
        J2 = .latitud + (A1 / 2)
        J4 = (3 * J2 + A2) / 4
        J6 = (5 * J4 + A2 * cosQuadrat(.latitud)) / 3
        alfa = (3 / 4) * elipsDesti.segonaExcQuadrat
        beta = (5 / 3) * Math.Pow(alfa, 2)
        gamma = (35 / 27) * Math.Pow(alfa, 3)
        Bphi = 0.9996 * elipsDesti.c * (.latitud - alfa * J2 + beta * J4 - gamma *
J6)

    End With

    pUTM.easting = Math.Floor(xi * v * (1 + zeta / 3) + 500000)

    If coordGeodD.latitud < 0 Then

        pUTM.northing += 10000000
        pUTM.hemisferi = "S"

    Else

        pUTM.hemisferi = "N"

    End If

    pUTM.northing = Math.Floor(eta * v * (1 + zeta) + Bphi)

    Return pUTM

End Function

Private Function cosQuadrat(ByVal valor As Double) As Double

    Return Math.Cos(valor) * Math.Cos(valor)

End Function

Private Function sinQuadrat(ByVal valor As Double) As Double

    Return Math.Sin(valor) * Math.Sin(valor)

End Function

End Class

```

2.3. costats.vb

```

Friend Class costats

    'Els passos indiquen quant varia la coordenada UTM per cada pixel del mapa
    'Són els valors que utilitzarem per a situar qualsevol punt en coordenades pantalla
    'a la seva representació en coordenades UTM
    Public pasXDalt As Double
    Public pasYDalt As Double
    Public pasXBaix As Double
    Public pasYBaix As Double
    Public pasXEsquerra As Double
    Public pasYEsquerra As Double
    Public pasXDreta As Double
    Public pasYDreta As Double

    'Són les coordenades UTM en el punt d'origen de les coordenades pantalla
    'Corresponen al punt (0,0) (superior esquerra) del mapa
    Public origenUTM As coordUTM

    'Punts de la recta que considerem
    Public origen1 As Point      'Punt d'origen de la primera recta
    Public desti1 As Point

```

```

Public origen2 As Point      'Punt d'origen de la segona recta
Public desti2 As Point
Public origen3 As Point      'Punt d'origen de la recta que s'uneix amb un punt de
tall
Public desti3 As Point

'Punts de tall amb el costat corresponent (coordenades pantalla)
Public puntTall1 As Point 'privat
Public puntTall2 As Point 'privat

'Punts de tall amb el costat corresponent (coordenades UTM)
Public pUTM1 As New coordUTM
Public pUTM2 As New coordUTM

'Array dels punts calibrats un cops ordenats
Public arrayCoordOrdenat() As puntCalibratge 'serà privat

Public Sub New()

    numPuntsCalibratge = 3
    'ReDim arrayPuntsCalibratge(numPuntsCalibratge - 1)
    ReDim arrayPuntsDatumActual(numPuntsCalibratge - 1)
    ReDim arrayCoordOrdenat(numPuntsCalibratge - 1)

End Sub

'Procediment que ens ordena 3 punts de manera que podem treballar amb ells
Private Sub ordenarPunts(ByVal datum As Integer)

    'Ordenem els punts de la manera que ens interessa:
    'aCO(0) = punt més a l'esquerre (amb un x coordPantalla menor)
    'aCO(1) = punt a la dreta més amunt (amb un y coordPantalla menor)
    'aCO(2) = punt a la dreta més abaix (amb un y coordPantalla major)

    Dim xMenor As Integer = 1000000
    Dim punt1Seleccionat As Integer
    Dim yMenor As Integer = 1000000
    Dim punt2Seleccionat As Integer

    If datum = 0 Then

        'S'ordena el datum actual
        For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
            If arrayPuntsDatumActual(i).coordPantalla.X < xMenor Then
                arrayCoordOrdenat(0) = arrayPuntsDatumActual(i)
                xMenor = arrayPuntsDatumActual(i).coordPantalla.X
                punt1Seleccionat = i
            End If
        Next

        For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
            If punt1Seleccionat <> i Then
                If arrayPuntsDatumActual(i).coordPantalla.Y < yMenor Then
                    arrayCoordOrdenat(1) = arrayPuntsDatumActual(i)
                    yMenor = arrayPuntsDatumActual(i).coordPantalla.Y
                    punt2Seleccionat = i
                End If
            End If
        Next

        For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
            If punt1Seleccionat <> i AndAlso punt2Seleccionat <> i Then
                arrayCoordOrdenat(2) = arrayPuntsDatumActual(i)
            End If
        Next

    Else

```

```

'S'ordena el datum de calibratge
For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
    If arrayPuntsCalibratge(i).coordPantalla.X < xMenor Then
        arrayCoordOrdenat(0) = arrayPuntsCalibratge(i)
        xMenor = arrayPuntsCalibratge(i).coordPantalla.X
        punt1Seleccionat = i
    End If
Next

For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
    If punt1Seleccionat <> i Then
        If arrayPuntsCalibratge(i).coordPantalla.Y < yMenor Then
            arrayCoordOrdenat(1) = arrayPuntsCalibratge(i)
            yMenor = arrayPuntsCalibratge(i).coordPantalla.Y
            punt2Seleccionat = i
        End If
    End If
Next

For i As Integer = 0 To numPuntsCalibratge - 1 Step 1
    If punt1Seleccionat <> i AndAlso punt2Seleccionat <> i Then
        arrayCoordOrdenat(2) = arrayPuntsCalibratge(i)
    End If
Next

End If

End Sub

Private Sub puntsTall(ByVal costat As String)

    Dim A1, B1, C1, A2, B2, C2 As Integer

    'Seleccionem el vector corresponent segons quin costat estem calculant
    Select Case costat

        Case "dalt"
            'La primera recta passa pel punt y més gran i per l'esquerre
            origen1 = arrayCoordOrdenat(2).coordPantalla
            desti1 = arrayCoordOrdenat(0).coordPantalla

            'La segona recta passa pel punt de l'esquerre i y més petit
            origen2 = desti1
            desti2 = arrayCoordOrdenat(1).coordPantalla

        Case "esquerra"
            'La primera recta passa per l'y més petit i l'esquerre
            origen1 = arrayCoordOrdenat(1).coordPantalla
            desti1 = arrayCoordOrdenat(0).coordPantalla

            'La segona recta passa per l'y més gran i el punt esquerre
            origen2 = arrayCoordOrdenat(2).coordPantalla
            desti2 = desti1

    End Select

    'Calculem els paràmetres que defineixen les rectes que passen per aquests punts
    A1 = desti1.Y - origen1.Y
    B1 = -desti1.X + origen1.X
    C1 = (desti1.X - origen1.X) * origen1.Y - (desti1.Y - origen1.Y) * origen1.X

    A2 = desti2.Y - origen2.Y
    B2 = -desti2.X + origen2.X
    C2 = (desti2.X - origen2.X) * origen2.Y - (desti2.Y - origen2.Y) * origen2.X

    'D'acord amb els paràmetres obtinguts, i tenint en compte el costat
    'apliquem les fórmules que ens retornen els punts de tall
    Select Case costat

        Case "dalt"
            puntTall1.X = -C1 / A1
            puntTall1.Y = 0

            puntTall2.X = -C2 / A2
            puntTall2.Y = 0
    End Select

```

```

Case "esquerra"
    puntTall1.X = 0
    puntTall1.Y = -C1 / B1

    puntTall2.X = 0
    puntTall2.Y = -C2 / B2

End Select

End Sub

'Procediment que calcula quines són les coordenades UTM del punts de tall del
'costat donat, a partir dels punts de tall en coordenades pantalla
Private Sub puntsTallUTM(ByVal costat As String)

    Dim i, j As Integer                'Indexos dels punts de l'array ordenat
    Dim dist1 As Double                'Distància entre els dos punts ordenats
    Dim dist2 As Double                'Distància entre un punt i el punt de tall

    'Calculem les distàncies per saber la proporcionalitat a l'hora
    'de trobar la coordenada UTM corresponent al punt de tall
    dist1 = calcularDistancia(origen1, desti1)

    'Només ens cal definir el 3r punt, ja que els altres 2 encara
    'els tenim guardats de la funció de puntsTall
    Select Case costat

        Case "dalt"
            'La primera recta que talla amb la pantalla passa pel
            'punt esquerre pel puntTall1
            origen3 = arrayCoordOrdenat(0).coordPantalla
            desti3 = puntTall1
            i = 0
            j = 2

        Case "esquerra"
            'La primera recta que talla passa pel punt esquerre i el puntTall1
            origen3 = arrayCoordOrdenat(0).coordPantalla
            desti3 = puntTall1
            i = 0
            j = 1

    End Select

    'Calculem la distància entre el punt més proper al límit del mapa i el punt
    'de tall corresponent
    dist2 = calcularDistancia(origen3, desti3)

    'Ara ja podem calcular les coordenades UTM d'aquest punt, ja que coneixem
    'la proporcionalitat existent
    pUTM1.northing = ((arrayCoordOrdenat(i).northing -
arrayCoordOrdenat(j).northing) * dist2 / dist1) + arrayCoordOrdenat(i).northing
    pUTM1.easting = ((arrayCoordOrdenat(i).easting - arrayCoordOrdenat(j).easting) *
dist2 / dist1) + arrayCoordOrdenat(i).easting
    pUTM1.zona = arrayCoordOrdenat(i).zona
    pUTM1.hemisferi = arrayCoordOrdenat(i).hemisferi

    'Repetim el procés per al segon punt de tall
    dist1 = calcularDistancia(origen2, desti2)

    Select Case costat
        Case "dalt"
            'La segona recta passa pel punt y més petit i puntTall2
            origen3 = arrayCoordOrdenat(1).coordPantalla
            desti3 = puntTall2
            i = 1
            j = 0

        Case "esquerra"
            'La segona recta passa per l'esquerra i puntTall2
            origen3 = arrayCoordOrdenat(0).coordPantalla
            desti3 = puntTall2
            i = 0
            j = 2

    End Select

```

```

dist2 = calcularDistancia(origen3, desti3)

pUTM2.northing = ((arrayCoordOrdenat(i).northing -
arrayCoordOrdenat(j).northing) * dist2 / dist1) + arrayCoordOrdenat(i).northing
pUTM2.easting = ((arrayCoordOrdenat(i).easting - arrayCoordOrdenat(j).easting) *
dist2 / dist1) + arrayCoordOrdenat(i).easting
pUTM2.zona = arrayCoordOrdenat(i).zona
pUTM2.hemisferi = arrayCoordOrdenat(i).hemisferi

End Sub

'Funció que retorna la distància euclídia entre dos punts (coordenades pantalla)
Public Function calcularDistancia(ByVal origen As Point, ByVal desti As Point) As
Double

Return Math.Sqrt((Math.Pow((desti.X.ToString - origen.X.ToString), 2)) + _
(Math.Pow((desti.Y.ToString - origen.Y.ToString),
2)))

End Function

'Procediment que calcula el pas (número de coordenades UTM que representa cada
'píxel al mapa) segons els valors del punt de tall d'aquest costat
Private Sub calculPas(ByVal costat As String)

Select Case costat
Case "dalt"
pasXDalt = (pUTM2.easting - pUTM1.easting) / (puntTall2.X.ToString -
puntTall1.X.ToString)
'MsgBox("pasxdalt: " & pasXDalt)

pasYDalt = (pUTM2.northing - pUTM1.northing) / (puntTall2.X.ToString -
puntTall1.X.ToString)
'MsgBox("pasydalt: " & pasYDalt)

Case "esquerra"
pasXEsquerra = (pUTM2.easting - pUTM1.easting) / (puntTall2.Y.ToString -
puntTall1.Y.ToString)
'MsgBox("pasxesquerra: " & pasXEsquerra)

pasYESquerra = (pUTM2.northing - pUTM1.northing) / (puntTall2.Y.ToString -
puntTall1.Y.ToString)
'MsgBox("pasyesquerra: " & pasYESquerra)
End Select
End Sub

'Sabent els punts de tall de la part superior de la pantalla i els seus pasos,
'podem determinar quina és la coordenada UTM origen del mapa.
Private Sub origenCoordenadesUTM()

origenUTM.zona = pUTM1.zona
origenUTM.hemisferi = pUTM1.hemisferi

If puntTall1.X > 0 Then
origenUTM.easting = pUTM1.easting - pasXDalt * puntTall1.X
origenUTM.northing = pUTM1.northing - pasYDalt * puntTall1.X
Else
origenUTM.easting = pUTM1.easting + pasXDalt * Math.Abs(puntTall1.X)
origenUTM.northing = pUTM1.northing + pasYDalt * Math.Abs(puntTall1.X)
End If

End Sub

'Procediment que calcula els passos (quina variació del northing o l'easting
'correspon a la variació d'un píxel, a més de calcular l'origen de coordenades UTM
Public Sub calcularPasos(ByVal datum As Integer)

'Primer ordenem els punts
'Si es rep un 0 s'ordenen els punts del datum actual
'Si es rep un 1 s'ordenen els del datum del calibratge
ordenarPunts(datum)

'Busquem els passos de cadascun dels costats de la pantalla:
'Volem saber quantes coordenades UTM corresponen a cada píxel

```

```

'Cas: part superior del mapa
puntsTall("dalt")
puntsTallUTM("dalt")
calculPas("dalt")
origenCoordenadesUTM()

'Cas: part esquerra del mapa
puntsTall("esquerra")
puntsTallUTM("esquerra")
calculPas("esquerra")

'Comprovem que algun dels costats no sigui 0 -> es realitza malament el mapejat
If pasXDalt = 0 Then
    pasXDalt = 1.0E-18
End If
If pasYDalt = 0 Then
    pasYDalt = 1.0E-18
End If

If pasXEsquerra = 0 Then
    pasXEsquerra = 1.0E-18
End If
If pasYEsquerra = 0 Then
    pasYEsquerra = 1.0E-18
End If

End Sub

End Class

```

2.4. NmeaInterpreter.vb

```

Public Class NmeaInterpreter

'Variables on es guarda informació per llegir-la i dibuixar
Friend coordGeodRebuda As New coordenadaGeodesica

'S'activa quan es de posició
Public Event PositionReceived(ByVal latitud As Double, ByVal longitud As Double)

'Processa informació rebuda del GPS
Public Function Parse(ByVal sentence As String) As Boolean

'Es mira el tipus de frase
Select Case GetWords(sentence) (0)

    Case "$GPRMC"

        'RMC
        Return ParseGPRMC(sentence)

    Case Else

        'No és una frase reconeguda
        Return False

    End Select

End Function

'Separa la frase segons els camps
Public Function GetWords(ByVal sentence As String) As String()

    Return sentence.Split(",")

End Function

'Analitza la frase GPRMC
Public Function ParseGPRMC(ByVal sentence As String) As Boolean

```

```

'Es divideix per camps
Dim Words() As String = GetWords(sentence)

'Conté prou dades?
If Words(3) <> "" AndAlso Words(4) <> "" AndAlso Words(5) <> "" AndAlso Words(6)
<> "" Then

    'Extreure latitud i longitud
    Dim latitud As Double
    Dim longitud As Double

    Dim graus As Double = Words(3).Substring(0, 2).ToString
    Dim minuts As Double = Words(3).Substring(2).ToString

    latitud = graus + (minuts.ToString / 10000) / 60

    If Words(4) = "S" Then

        latitud = -latitud

    End If

    graus = Words(5).Substring(0, 3).ToString
    minuts = Words(5).Substring(3).ToString

    longitud = graus + (minuts.ToString / 10000) / 60

    If Words(6) = "W" Then

        longitud = -longitud

    End If

    RaiseEvent PositionReceived(latitud, longitud)

End If

'Sentència vàlida
Return True

End Function

'Comprova si el checksum és correcte
Public Function IsValid(ByVal sentence As String) As Boolean

    'Es comparen els dos checksums
    Return sentence.Substring(sentence.IndexOf("**") + 1) = GetChecksum(sentence)

End Function

'Calcula el checksum d'una frase
Public Function GetChecksum(ByVal sentence As String) As String

    Dim Character As Char
    Dim Checksum As Integer

    'Es miren tots els caràcters
    For Each Character In sentence

        Select Case Character
            Case "$"c
                'Si és un dòlar s'ignora

            Case "*"c
                'Si és un asterisc, ja s'ha acabat
                Exit For

            Case Else

                'Es calcula el checksum acumulat
                If Checksum = 0 Then

                    Checksum = Convert.ToByte(Character)

                Else

```

```
Checksum = Checksum Xor Convert.ToByte(Character)

End If

End Select

Next

'Retorna el checksum en format hexadecimal en 2 caracters
Return Checksum.ToString("X2")

End Function

Private Sub NmeaInterpreter_PositionReceived(ByVal latitud As Double, ByVal longitud
As Double) Handles Me.PositionReceived

    coordGeodRebuda.latitud = latitud
    coordGeodRebuda.longitud = longitud

End Sub

End Class
```