```vb
Imports System.IO
Imports System.Data.oledb
Public Class Form1
    Public Ofd As New OpenFileDialog
    Dim miofile As StreamReader
    Dim a As Double 'x minima
    Dim b As Double 'x massima
    Dim c As Double 'y minima
    Dim d As Double 'y massima
    Dim ymin, ymax As Double
    Dim gr As Graphics
    Dim FlagFunc As Boolean
    Dim integ As Integrale


    Public Function cx(ByVal x As Double) As Integer
        Dim cox As Integer
        Try
            'a = txtXMin.Text
            'b = txtXMax.Text
            cox = (((x - a)) / ((b - a))) * Panel1.Width 'calcolo l'ascissa sul pannello
            Return cox
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Function

    Public Function cy(ByVal y As Double) As Integer
        Dim coy As Integer
        coy = (1 - ((((y - c)) / ((d - c))))) * Panel1.Height 'calcolo l'ordinata sul pannello
        Return coy
    End Function

    Function f(ByVal x As Double) As Double
        Dim ya As Double
        Dim media As Double
        Dim scarto As Double
        If txtMedia.Text <> "" And txtScarto.Text <> "" And RadioButton1.Checked = True Then
            media = txtMedia.Text
            scarto = txtScarto.Text
            ya = (1 / scarto * Math.Sqrt(2 * 3.14159265)) * (2.718281828 ^ -((x - media) ^ 2) / 2 * (scarto ^ 2)) 'Funzione
        ElseIf RadioButton2.Checked = True And lblS.Text <> "" And lblM.Text <> "" Then
            media = lblM.Text
            scarto = lblS.Text
            ya = (1 / scarto * Math.Sqrt(2 * 3.14159265)) * (2.718281828 ^ -((x - media) ^ 2) / 2 * (scarto ^ 2)) 'Funzione
        End If
        Return ya
    End Function

    Sub funzione()
```

```
        Dim x As Double
        Dim ya As Single
        gr = Panel1.CreateGraphics
        Dim incr As Double
        FlagFunc = True
        incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2)) 'dimensione incremento dell'indice
        For x = (a) To (b) Step incr 'da a fino a b incrementandosi di incr
            Try
                ya = f(x) 'ritorna il valore della funzione su questa x
                'gr.DrawEllipse(Pens.Red, cx(x), cy(ya), 1, 1)
                gr.DrawLine(Pens.Blue, cx(x), (cy(ya)), cx(x + incr), cy(f(x + incr))) 'disegna una linea da x,y correnti a x,y del ciclo successivo
            Catch ex As Exception

            End Try
        Next
        creaReticolo()
    End Sub

    Sub rangeY()
        Dim x, min, max As Double
        Dim ya As Single
        Dim incr As Double
        Dim flag1, flag2 As Boolean
        flag1 = False
        flag2 = False
        Try
            incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2))
            min = -0.0001 'massimo e minimo iniziali molto vicini allo 0
            max = 0.0001
            a = txtXMin.Text
            b = txtXMax.Text
            If (a < b) Then
                For x = a To b Step 1
                    ya = f(x) 'ya assume il valore della funzione nell'ascissa x
                    If (ya < min) And (ya > (-100)) Then 'se ya e' piu' piccola di min  e piu' grande del tetto minimo definito come -100
                        min = ya 'il minimo diventa ya
                        flag1 = True
                    End If
                    If (ya > max) And (ya < 100) Then 'se ya e' piu' grande di max  e piu' piccolo del tetto massimo definito come 100
                        max = ya 'il massimo diventa max
                        flag2 = True
                    End If
                Next
                c = (min - 2) 'diminuisco il minimo ottenuto di 2 e aumento il massimo di 2 per evitare di avere la funzione attaccata a punti non compresi o compresi al pelo
                d = (max + 2)
                If flag1 = False Then 'se non ho mai cambiato il minimo
                    min = ya 'min diventa l'ultima ya trovata che sara' > 0
                    For x = a To b Step 1
                        ya = f(x) 'ya assume il valore della variabile dipendente di x
                        If (ya < min) Then 'se ya e' minore dell'ultimo min posto (l'ultima ya in partenza)
                            min = ya 'allora min diventa la corrente ya
                        End If
                    Next
                    c = min - 2 'la c (valore minimo dell'ordinata della funzione) viene abbassato di 2 per rendere visibile la funzione
                End If
                If flag2 = False Then 'stesso lavoro di sopra, ma al contrario per ottenere l'ordinata massima nel caso l'ordinata massima sia minore di 0
                    max = ya
                    For x = a To b Step 1
                        ya = f(x)
                        If (ya > max) Then
                            max = ya
                        End If
                    Next
                    d = max + 2
                End If
            Else
                MsgBox("XMin deve essere minore di Xmax!")
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Sub assi()
        Dim gr As Graphics
        gr = Panel1.CreateGraphics
        gr.Clear(Color.White)
        Try
            If (b > 0) And (a < 0) Then
                gr.DrawLine(Pens.Black, cx(0), (cy(c)), cx(0), (cy(d))) 'asse y
            End If
            If (d > 0) And (c < 0) Then
                gr.DrawLine(Pens.Black, cx(a), cy(0), cx(b), cy(0)) 'asse x
            End If
            creaReticolo()
        Catch ex As Exception

        End Try
    End Sub

    Private Sub BtnDisegna_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnDisegna.Click
        rangeY()
        Button1.Visible = True
        If a < b Then
            assi() 'disegna assi
            funzione() 'disegna funzione
            Label3.Text = "Y Minima= " & Math.Truncate(c) 'mostra il valore dell'ord min troncato all'intero
            Label4.Text = "Y Massima= " & Math.Truncate(d) 'mostra il valore dell'ord max troncato all'intero
        End If

        If c > 0 Then 'casi in cui gli assi non sono visibili
            BtnVisAssi.Visible = True 'rendo visibile un pulsante per allargare la funzione e visualizzare gli assi
        End If
        If d < 0 Then
            BtnVisAssi.Visible = True
        End If
        If a > 0 Then
            BtnVisAssi.Visible = True
        End If
        If b < 0 Then
            BtnVisAssi.Visible = True
        End If

    End Sub

    Private Sub BtnVisAssi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnVisAssi.Click
        If c > 0 Then 'modifico il valore degli estremi per rendere visibili gli assi e ridisegno la funzione
            c = -2
        End If
        If d < 0 Then
            d = 2
        End If
        If a > 0 Then
            a = -2
        End If
        If b < 0 Then
            b = 2
        End If


        assi()
        funzione()
        Label3.Text = "Y Minima= " & Math.Truncate(c) 'mostra il valore dell'ord min troncato all'intero
        Label4.Text = "Y Massima= " & Math.Truncate(d) 'mostra il valore dell'ord max troncato all'intero


        BtnVisAssi.Visible = False
    End Sub

    'Private Sub BtnIndietro_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    '    assi()
    '    funzione()
    '    Label3.Text = "Y Minima= " & c
    '    Label4.Text = "Y Massima= " & d
    'End Sub

    Public Sub creaReticolo()
        Dim i As Single
        Dim gr As Graphics
        gr = Panel1.CreateGraphics
        Dim incr As Double
        Dim incr2 As Double
        Dim incrY As Double
        incr = Math.Abs(b - a) / 10
        incr2 = cx(incr) - cx(0)
        For i = incr2 To Panel1.Width Step incr2
            Try
                If (a < 0) And (b > 0) Then
                    gr.DrawLine(Pens.Fuchsia, cx(0) + i, 0, cx(0) + i, Panel1.Height)
                    gr.DrawLine(Pens.Fuchsia, cx(0) - i, 0, cx(0) - i, Panel1.Height)
                Else
                    gr.DrawLine(Pens.Fuchsia, cx(a) + i, 0, cx(a) + i, Panel1.Height)
                End If
            Catch ex As Exception

            End Try
        Next
        For i = incr2 To Panel1.Height Step incr2
            Try
                If (c < 0) And (d > 0) Then
                    gr.DrawLine(Pens.Fuchsia, 0, (cy(0) + i), Panel1.Width, (cy(0) + i))
                    gr.DrawLine(Pens.Fuchsia, 0, (cy(0) - i), Panel1.Width, (cy(0) - i))
                Else
                    gr.DrawLine(Pens.Fuchsia, 0, (cy(c) + i), Panel1.Width, (cy(c) + i))
                End If
            Catch ex As Exception

            End Try
        Next
        lblQW.Text = "Larghezza Quadrato=" & Math.Round(incr, 3)
        incrY = (Math.Abs(d - c) / (Panel1.Height / incr2))
        lblQH.Text = "Altezza Quadrato=" & Math.Round(incrY, 3)
    End Sub


    Private Sub BtnRI_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnRI.Click
        Dim baseRett As Double
        Dim i As Integer
        Dim altRett1 As Double
        Dim altRett2 As Double
        Dim altRett As Double
        Dim b1, b2 As Single
        Dim baseP As Single
        Dim altP As Single
        Dim n As Integer
        If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
            integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
            LblRI.Text = "Area Rett Inf=" & Math.Round(integ.rettangoliInf(a, b), 3)
            n = TxtN.Text
            baseRett = (b - a) / n
            For i = 1 To n
                Try
                    b1 = CSng(a + (baseRett * (i - 1)))
                    b2 = CSng(a + (baseRett * (i)))
                    altRett1 = f(a + (baseRett * (i - 1)))
                    altRett2 = f(a + (baseRett * i))
                    baseP = cx(baseRett) - cx(0)
                    If altRett1 < altRett2 Then
                        altRett = altRett1
                    Else
                        altRett = altRett2
                    End If
                    altP = Math.Abs(cy(altRett) - cy(0))
                    If ((f(altRett)) < 0) Then
                        gr.DrawRectangle(Pens.Blue, cx(b1), cy(0), baseP, altP)
                    Else
                        gr.DrawRectangle(Pens.Blue, cx(b1), cy(altRett), baseP, altP)
                    End If
                Catch ex As Exception

                End Try
            Next
        End If
    End Sub

    Private Sub BtnRS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnRS.Click
        Dim baseRett As Double
        Dim i As Integer
        Dim altRett1 As Double
        Dim altRett2 As Double
        Dim altRett As Double
        Dim b1, b2 As Single
        Dim baseP As Single
        Dim altP As Single
        Dim n As Integer
        If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
            integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
            LblRS.Text = "Area Rett Sup=" & Math.Round(integ.rettangoliSup(a, b), 3)
            n = TxtN.Text
            baseRett = (b - a) / n
            For i = 1 To n
```

```
                Try
                    b1 = CSng(a + (baseRett * (i - 1)))
                    b2 = CSng(a + (baseRett * (i)))
                    altRett1 = f(a + (baseRett * (i - 1)))
                    altRett2 = f(a + (baseRett * i))
                    baseP = cx(baseRett) - cx(0)
                    If altRett1 > altRett2 Then
                        altRett = altRett1
                    Else
                        altRett = altRett2
                    End If
                    altP = Math.Abs(cy(altRett) - cy(0))
                    If ((f(altRett)) < 0) Then
                        gr.DrawRectangle(Pens.Green, cx(b1), cy(0), baseP, altP)
                    Else
                        gr.DrawRectangle(Pens.Green, cx(b1), cy(altRett), baseP, altP)
                    End If
                Catch ex As Exception

                End Try
            Next
        End If
    End Sub

    Private Sub BtnT_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnT.Click
        Dim baseRett As Double
        Dim i As Integer
        Dim altRett1 As Double
        Dim altRett2 As Double
        Dim b1, b2 As Single
        Dim baseP As Single
        Dim altP1 As Single
        Dim altP2 As Single
        Dim n As Integer
        Dim punto1 As Point
        Dim punto2 As Point
        Dim punto3 As Point
        Dim punto4 As Point
        If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
            integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
            LblT.Text = "Area Trap=" & Math.Round(integ.trapezi(a, b), 3)
            n = TxtN.Text
            baseRett = (b - a) / n
            For i = 1 To n
                Try
                    b1 = CSng(a + (baseRett * (i - 1)))
                    b2 = CSng(a + (baseRett * (i)))
                    altRett1 = f(a + (baseRett * (i - 1)))
                    altRett2 = f(a + (baseRett * i))
                    baseP = cx(baseRett) - cx(0)
                    altP1 = Math.Abs(cy(altRett1) - cy(0))
                    altP2 = Math.Abs(cy(altRett2) - cy(0))
                    punto1.X = cx(b1)
                    punto1.Y = cy(0)
                    punto2.X = cx(b2)
                    punto2.Y = cy(0)
                    punto3.X = cx(b2)
                    punto3.Y = cy(altRett2)
                    punto4.X = cx(b1)
                    punto4.Y = cy(altRett1)

                    gr.DrawLine(Pens.Lavender, punto2.X, punto2.Y, punto3.X, punto3.Y)
                    gr.DrawLine(Pens.IndianRed, punto3.X, punto3.Y, punto4.X, punto4.Y)
                    gr.DrawLine(Pens.IndianRed, punto4.X, punto4.Y, punto1.X, punto1.Y)
                Catch ex As Exception

                End Try
            Next
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim media As Double
        Dim scarto As Double
        If txtMedia.Text <> "" And txtScarto.Text <> "" Then
            media = txtMedia.Text
            scarto = txtScarto.Text
            If RadioButton1.Checked = True Then
                GaussianaStd.a = (a - media) / scarto
                GaussianaStd.b = (b - media) / scarto
            Else
                GaussianaStd.a = (a - CDbl(lblM.Text)) / CDbl(lblS.Text)
                GaussianaStd.b = (b - CDbl(lblM.Text)) / CDbl(lblS.Text)
            End If
            GaussianaStd.ShowDialog()
        Else
            media = 0
            scarto = 0
            MsgBox("Inserisci il valore di media e scarto prima!!")
        End If
    End Sub

    Private Sub Tmr_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
        If a < b Then
            assi() 'disegna assi
            funzione() 'disegna funzione
        End If
    End Sub



    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```vb
        DGV.AllowUserToAddRows = False
    End Sub

    Private Sub leggiMdb(ByVal s As String)
        Dim i As Integer
        Dim conn As New OleDb.OleDbConnection()
        Dim nomi As New nomiTab
        Dim med, var As Double
        i = nomi.RitornaNomi(DGV, CMB, s)
        If (i <= 1) Then
            Dim cmd As OleDbDataAdapter
            Dim DS As New DataSet
            conn = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" & "Data Source=" & s)
            cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Items(0) & "]", conn)
            cmd.Fill(DS, "tabella")
            DGV.DataSource = DS.Tables(0)
            conn.Close()
        Else
            CMB.Visible = True
            DGV.Visible = False
        End If
        DGV.AllowUserToAddRows = False
        mediaVarMdb(DGV, med, var)
        var = Math.Sqrt(var)
        lblM.Text = Math.Round(med, 3)
        lblS.Text = Math.Round(var, 3)
    End Sub

    Public Sub leggiCsv(ByVal s As String)
        Dim strV As String()
        Dim y, i, colCount As Integer
        Dim Flag As Boolean = False
        Dim med, var As Double
        colCount = 1
        Dim a As String = ""
        Try
            miofile = File.OpenText(s)
            While miofile.Peek <> -1
                a = miofile.ReadLine
                strV = Split(a, ",")
                For i = 0 To strV.Length - 1
                    If y = 0 Then
                        colCount += 1
                        DGV.Columns.Add(strV(i), strV(i))
                    Else
                        If y <> 0 And colCount <= strV.Length Then
                            MsgBox("Troppi elementi nella riga" & y)
                            Flag = True
                            Exit For
                        End If
                        If y <> 0 And i = 0 Then
                            DGV.Rows.Add()
                        End If
                        If y <> 0 And colCount > strV.Length Then
```

```
                            DGV.Item(i, (y - 1)).Value = strV(i)
                        End If
                    End If
                Next
                If Flag = True Then
                    Exit While
                End If
                y += 1
            End While
            DGV.AllowUserToAddRows = False
            miofile.Close()
            mediaVar(DGV, med, var)
            var = Math.Sqrt(var)
            lblM.Text = Math.Round(med, 3)
            lblS.Text = Math.Round(var, 3)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Public Sub leggiXls(ByVal s As String)
        Dim DS As New DataSet
        Dim i As Integer
        Dim nomi As New nomiTab
        Dim med, var As Double
        Dim cmd As OleDb.OleDbDataAdapter
        i = nomi.RitornaNomi(DGV, CMB, s)
        If (i <= 1) Then
            Dim conn As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" & "Data Source=" & s & ";" & "Extended Properties='Excel 8.0;HDR=YES;'")
            cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Items(0) & "]", conn)
            cmd.Fill(DS, "tabella")
            DGV.DataSource = DS.Tables(0)
            conn.Close()
        Else
            CMB.Visible = True
            DGV.Visible = False
        End If
        DGV.AllowUserToAddRows = False
        mediaVar(DGV, med, var)
        var = Math.Sqrt(var)
        lblM.Text = Math.Round(med, 3)
        lblS.Text = Math.Round(var, 3)
    End Sub

    Private Sub CMB_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CMB.SelectedIndexChanged
        Dim cmd As OleDbDataAdapter
        Dim DS As New DataSet
        If Strings.Right(Ofd.FileName, 3) = "mdb" Then
            Dim conn As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Ofd.FileName)
            cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Text & "]", conn)
            cmd.Fill(DS, "tabella").ToString()
            DGV.DataSource = DS.Tables(0)
            conn.Close()
        End If
        If Strings.Right(Ofd.FileName, 3) = "xls" Then
            Dim conn As New OleDbConnection
              ("Provider=Microsoft.Jet.OLEDB.4.0;" & "Data Source=" &
              Ofd.FileName & ";" & "Extended Properties='Excel 8.0;HDR=YES;'")
            cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Text & "]",
              conn)
            cmd.Fill(DS, "tabella").ToString()
            DGV.DataSource = DS.Tables(0)
            conn.Close()
        End If
        CMB.Visible = False
        DGV.Visible = False
        BtnDisegna.Visible = True
    End Sub

    Private Sub importa()
        DGV.DataSource = Nothing
        DGV.Columns.Clear()
        Ofd.InitialDirectory = "C:\Users\Seyriu\Desktop\ApriFileInDGV
          \ApriFileInDGV\bin\Debug"
        Ofd.Title = "Apri"
        Ofd.Filter = "Access Files|*.mdb|Excel Files|*.xls|Text Files|*.csv"
        Ofd.ShowDialog()
        If Strings.Right(Ofd.FileName, 3) = "mdb" Then
            Try
                leggiMdb(Ofd.FileName)
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        End If
        If Strings.Right(Ofd.FileName, 3) = "csv" Then
            Try
                leggiCsv(Ofd.FileName)
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        End If
        If Strings.Right(Ofd.FileName, 3) = "xls" Then
            Try
                leggiXls(Ofd.FileName)
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        End If
        If Strings.Right(Ofd.FileName, 3) = "csv" Then
            BtnDisegna.Visible = True
        End If
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
      ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
        BtnDisegna.Visible = True
        'Label6.Visible = True
        'Label7.Visible = True
        txtMedia.Visible = True
        txtScarto.Visible = True
        lblM.Visible = False
        lblS.Visible = False
        BtnApri.Visible = False
        CMB.Visible = False
        Comb.Visible = False
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
        BtnDisegna.Visible = False
        'Label6.Visible = False
        'Label7.Visible = False
        txtMedia.Visible = False
        txtScarto.Visible = False
        lblM.Visible = True
        lblS.Visible = True
        BtnApri.Visible = True
        'CMB.Visible = True
    End Sub

    Private Sub BtnApri_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnApri.Click
        importa()
        If Not Strings.Right(Ofd.FileName, 3) = "csv" Then
            CMB.Visible = True
            BtnDisegna.Visible = True
            Comb.Visible = True
        End If
    End Sub

    Private Sub mediaVarMdb(ByVal dgv As DataGridView, ByRef media As Double, ByRef varianza As Double)
        Dim i As Integer
        Dim somma, mediaS As Double

        For i = 0 To dgv.RowCount - 1
            somma += dgv.Item(2, i).Value
        Next
        media = somma / dgv.RowCount

        somma = 0
        For i = 0 To dgv.RowCount - 1
            somma += (dgv.Item(2, i).Value) ^ 2
        Next
        mediaS = somma / dgv.RowCount
        varianza = mediaS - (media ^ 2)
    End Sub

    Private Sub mediaVar(ByVal dgv As DataGridView, ByRef media As Double, ByRef varianza As Double)
        Dim i As Integer
        Dim somma, mediaS As Double
```

```vb
        For i = 0 To dgv.RowCount - 1
            somma += dgv.Item(1, i).Value
        Next
        media = somma / dgv.RowCount

        somma = 0
        For i = 0 To dgv.RowCount - 1
            somma += (dgv.Item(1, i).Value) ^ 2
        Next
        mediaS = somma / dgv.RowCount
        varianza = mediaS - (media ^ 2)
    End Sub

    Private Sub Comb_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Comb.Click
        Combinazioni.Show()
    End Sub
End Class

```