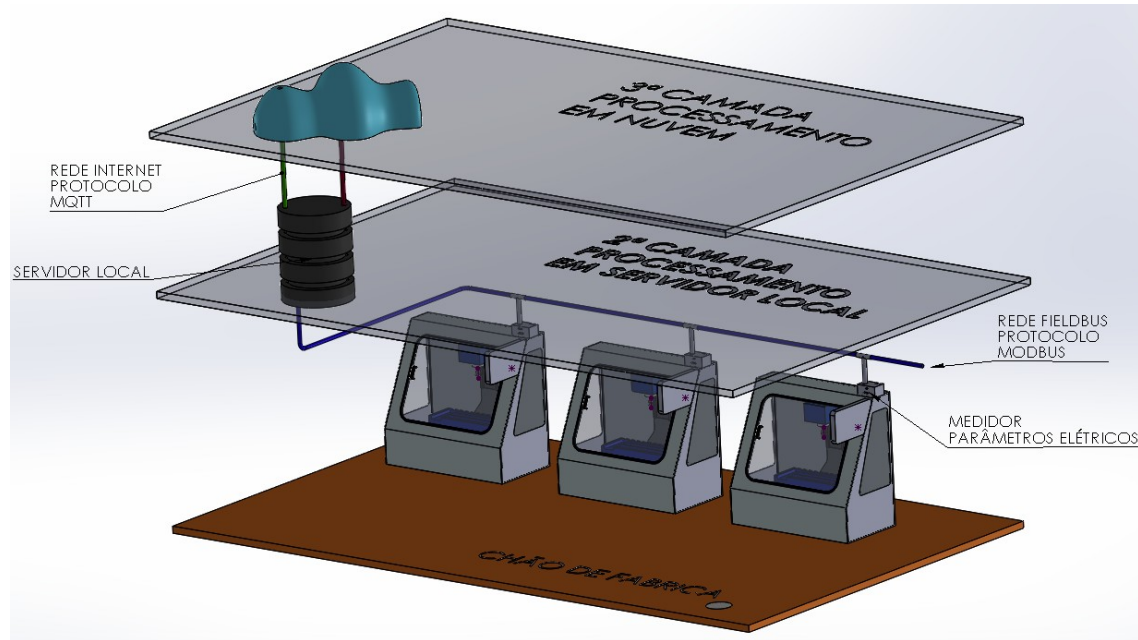
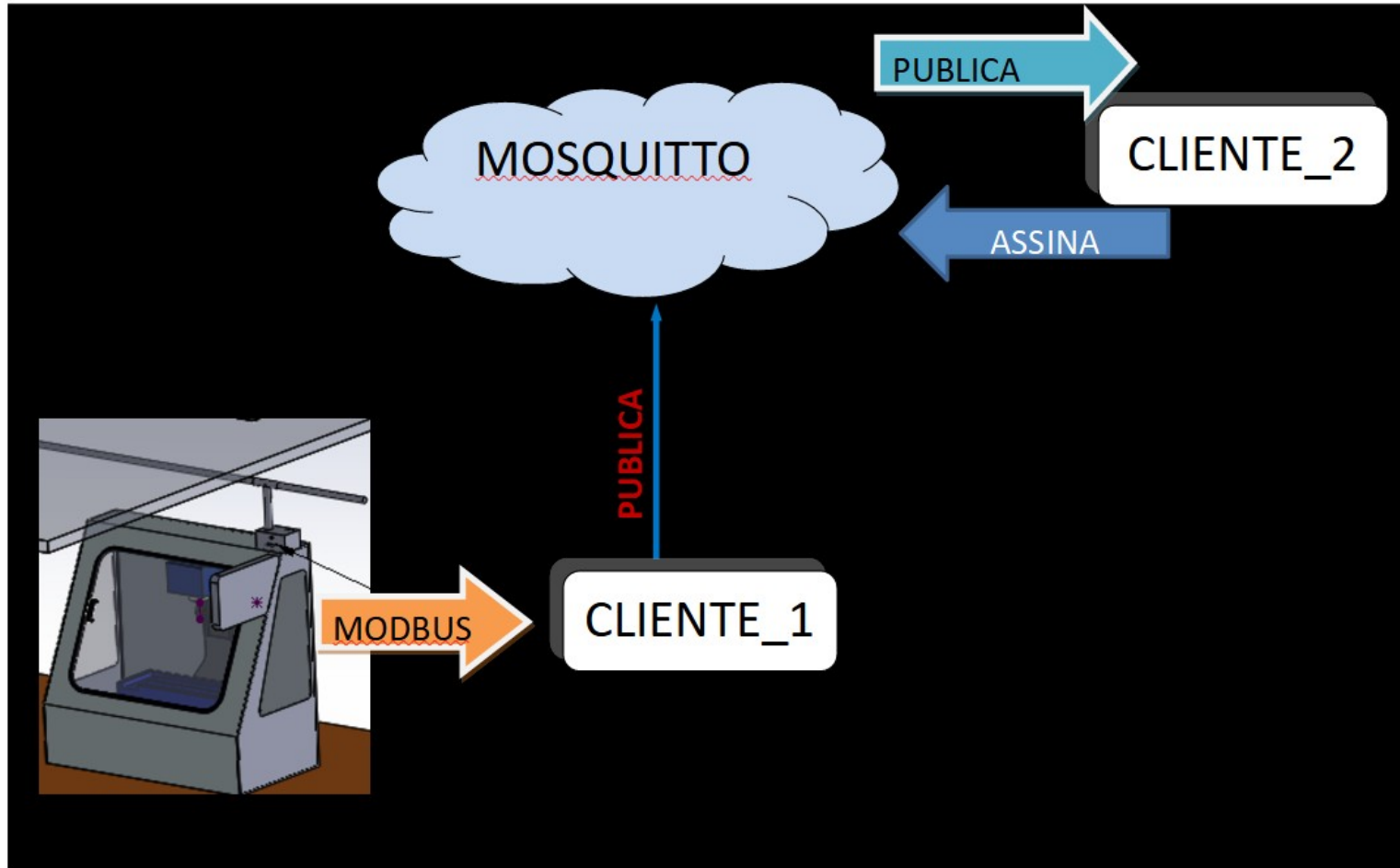


Implementação do projeto MPEU com MS-Visual Studio



projeto MPEU



Material de apoio

* Programando em Visual Basic .Net2010 – **introdução** – disponível em :
https://labusig.ufpr.br/wp-content/uploads/2024/02/apst_vb_2010_express.pdf

*Projeto completo do MPEU – disponível em :
<https://labusig.ufpr.br/wp-content/uploads/2024/03/aula3-parte1-2021t3.pdf>

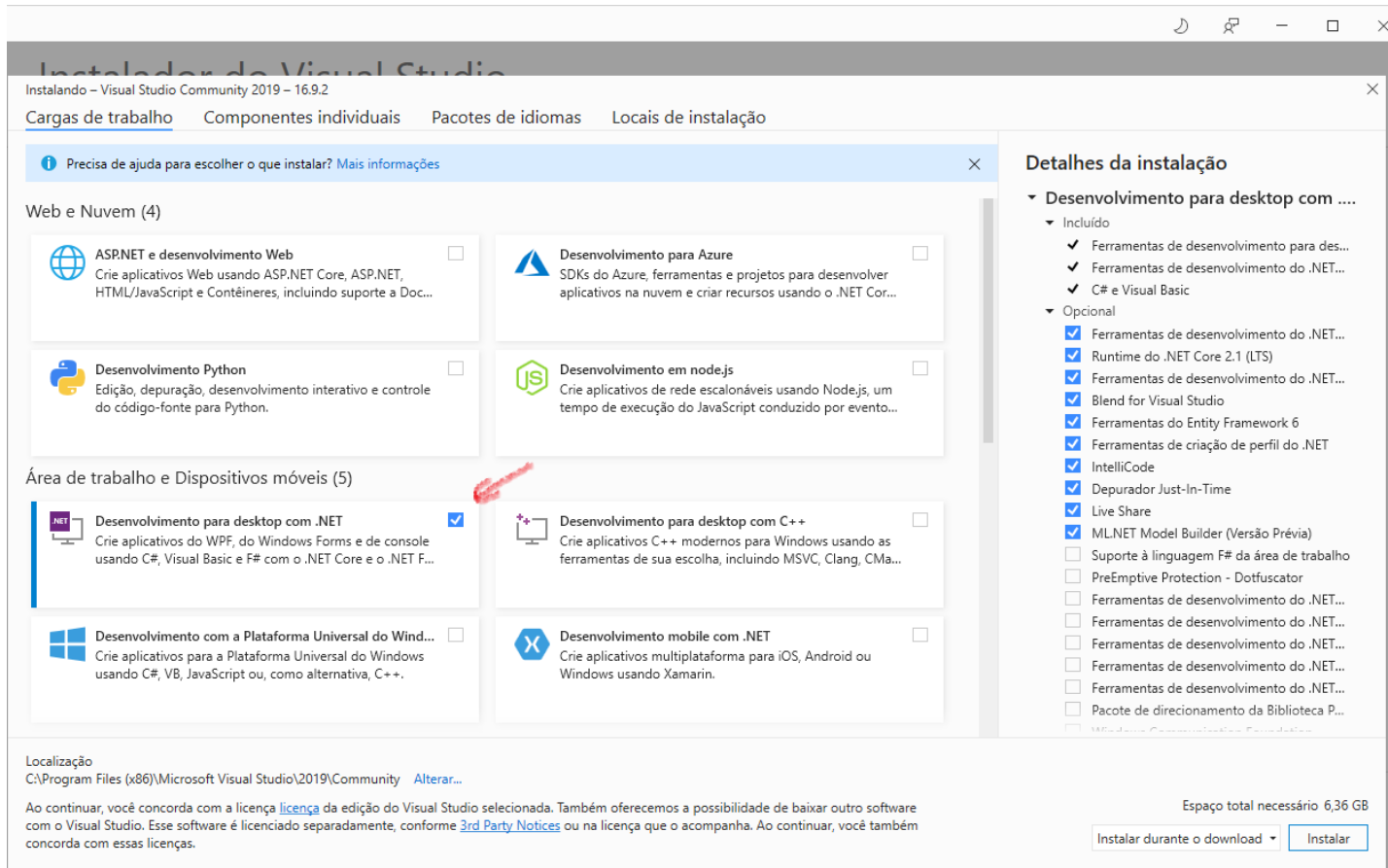
* estudo de caso envolvendo o emprego dessa tecnologia para o monitoramento em tempo real de uma linha de usinagem do tipo transfer rotativa – disponível em:
<http://dx.doi.org/10.1007/s11740-024-01284-0>

PROGRAMANDO EM VISUAL BASIC NO MS-VISUAL STUDIO

- Baseada em elementos gráficos (controles) e, atualmente, com recursos para programação orientada por objetos
- Possibilita o desenvolvimento de aplicativos para Windows com baixo custo e tempo
- Teve início com um IDE (Integrated Development Environment) dedicado: VB 3.0; VB 4.4; VB5.0 e VB6.0
- Em 2001, com o lançamento da plataforma .NET Framework , a Microsoft lançou o Visual Studio com sendo o novo IDE para programação em VB, C++, C# e outros.
- A partir de 2010 a Microsoft lançou um versão gratuita do Visual Studio e foi denominada com Visual Studio Express e atualmente é conhecida como Community e já está na versão 2022.

Instalação do visual studio

Versão 2019



A instalação pode ser completa – todos os recursos

Ou apenas do que for necessário, tal como indicado pela seta vermelha

Criando o primeiro projeto

Visual Studio 2019

Abrir recente

Para um acesso rápido, todos os projetos, as pastas ou os arquivos que forem abertos serão exibidos aqui.

Você pode fixar qualquer item usado com frequência para que ele fique sempre no topo da lista.

Introdução

habilitados para a nuvem



Clonar um repositório

Obter o código de um repositório online, como o GitHub ou o Azure DevOps



Abrir um projeto ou uma solução

Abrir um projeto local do Visual Studio ou arquivo .sln



Abrir uma pasta local

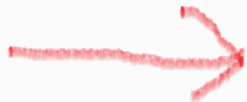
Navegar e editar o código dentro de qualquer pasta



Criar um projeto

Escolha um modelo de projeto com scaffolding de código para começar

[Continuar sem código →](#)



Criar um novo projeto

Modelos de projeto recentes

Uma lista de seus modelos acessados recentemente será exibida aqui.

Pesquisar modelos (Alt+S)



Todos os idiomas

Todas as plataformas

Todos os tipos de projeto



ASP.NET Core com React.js

Um modelo de projeto para a criação de um aplicativo ASP.NET Core com React.js

C# Linux macOS Windows Nuvem Serviço Web



ASP.NET Core com React.js e Redux

Um modelo de projeto para a criação de um aplicativo ASP.NET Core com React.js e Redux

C# Linux macOS Windows Nuvem Serviço Web



Aplicativo do Windows Forms

Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.

C# Windows Área de Trabalho



Aplicativo do Windows Forms

Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.

Visual Basic Windows Área de Trabalho



Aplicativo WPF

Um projeto para criar um Aplicativo WPF .NET Core

Voltar

Próximo

Selecione a opção “Aplicativo do Windows Forms”

Configurar seu novo projeto

Aplicativo do Windows Forms

Visual Basic

Windows

Área de Trabalho

Nome do projeto

Cliente2

Local

C:\Users\LABUSIG\Desktop\CLIENTE-MQTT\

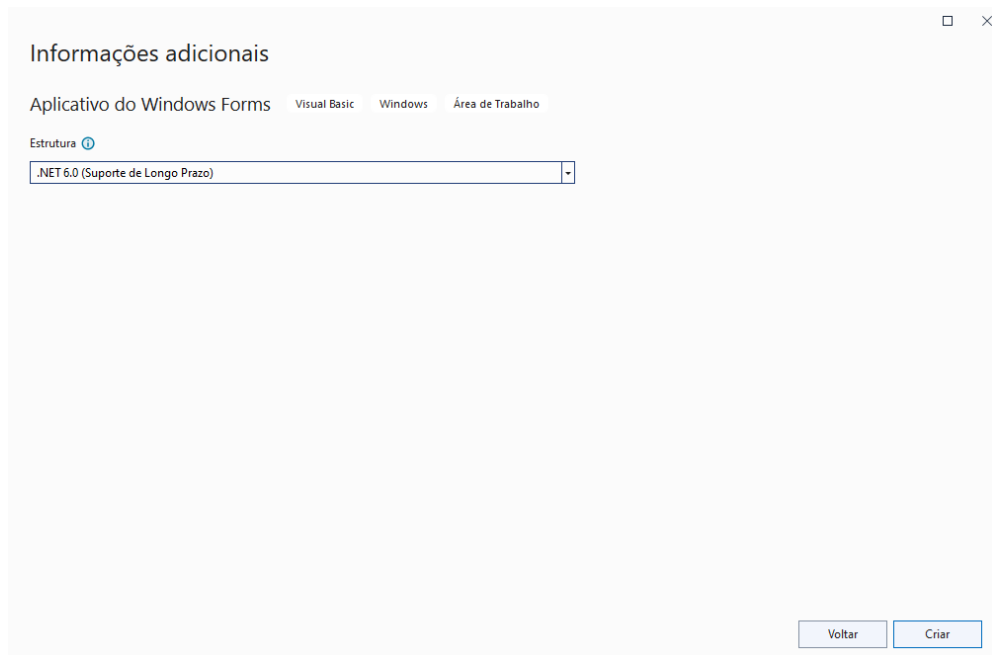
...

Nome da solução ⓘ

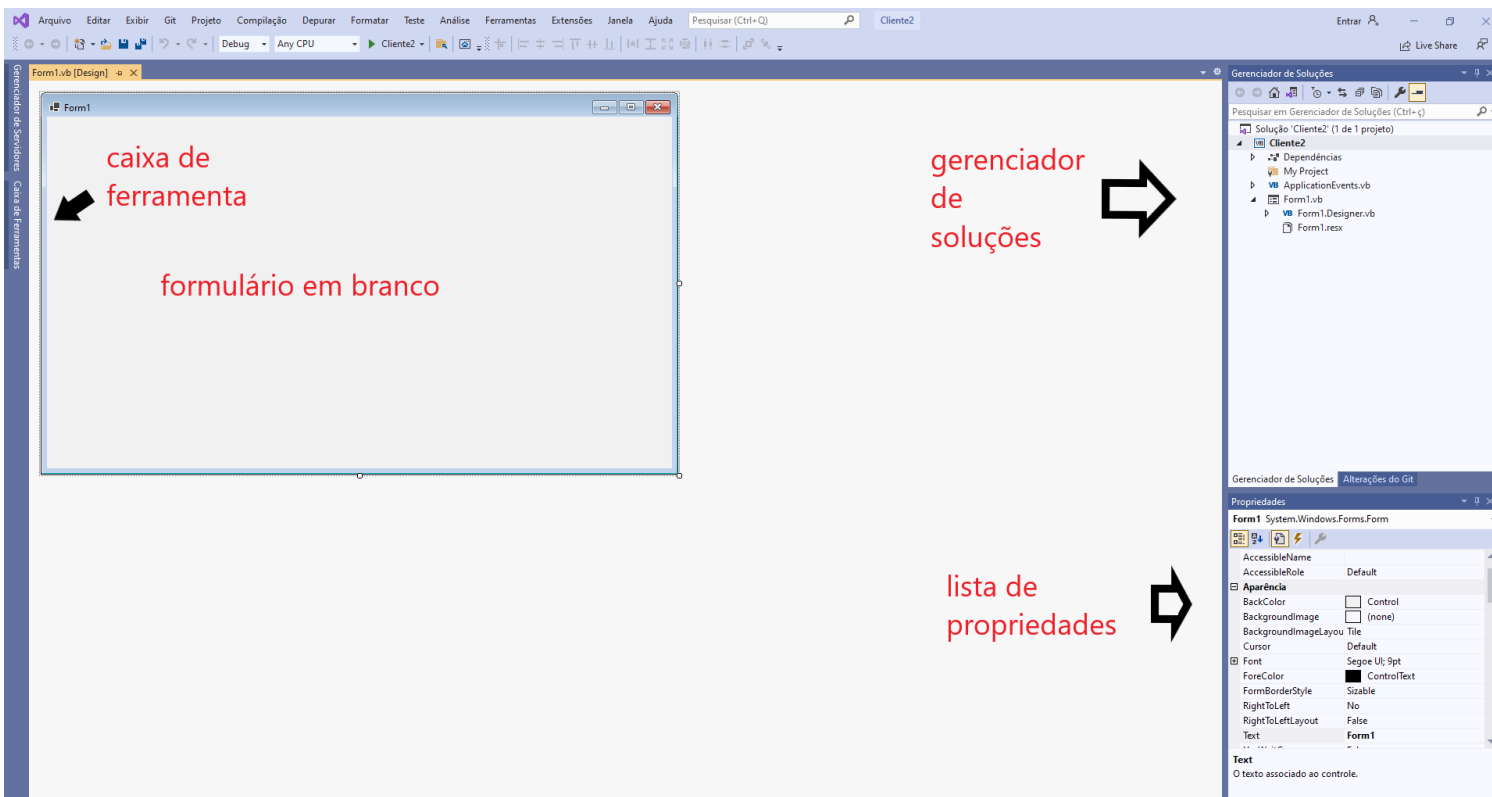
Cliente2

☐ Colocar a solução e o projeto no mesmo diretório

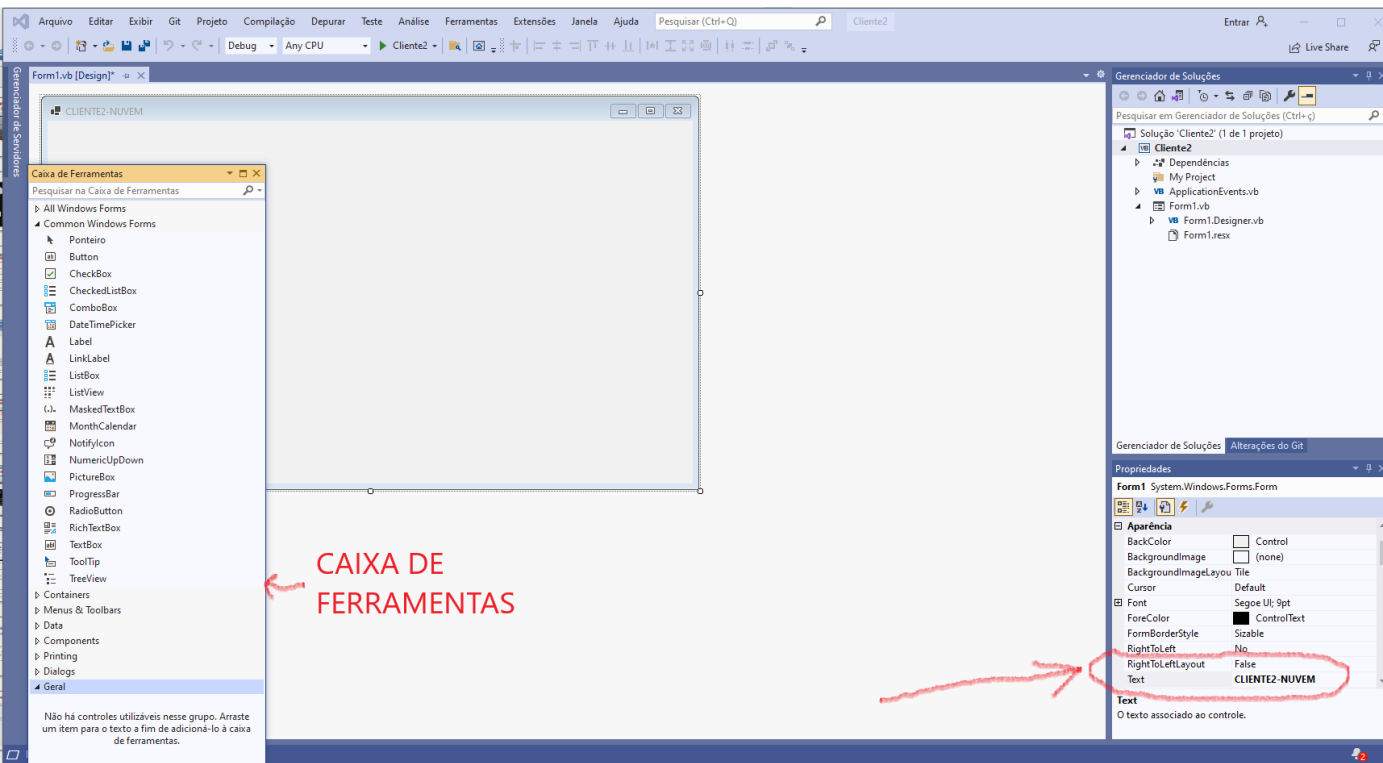
Dê um nome para o projeto e escolha um local para armazenagem dos arquivos que serão integrados pelo “gerenciador de soluções”



Escolha a estrutura .NET



Após acionar o botão “OK” aparece (automaticamente) um formulário vazio; o “Gerenciador de Soluções”; a “caixa de ferramentas”; a “lista de propriedades”;...

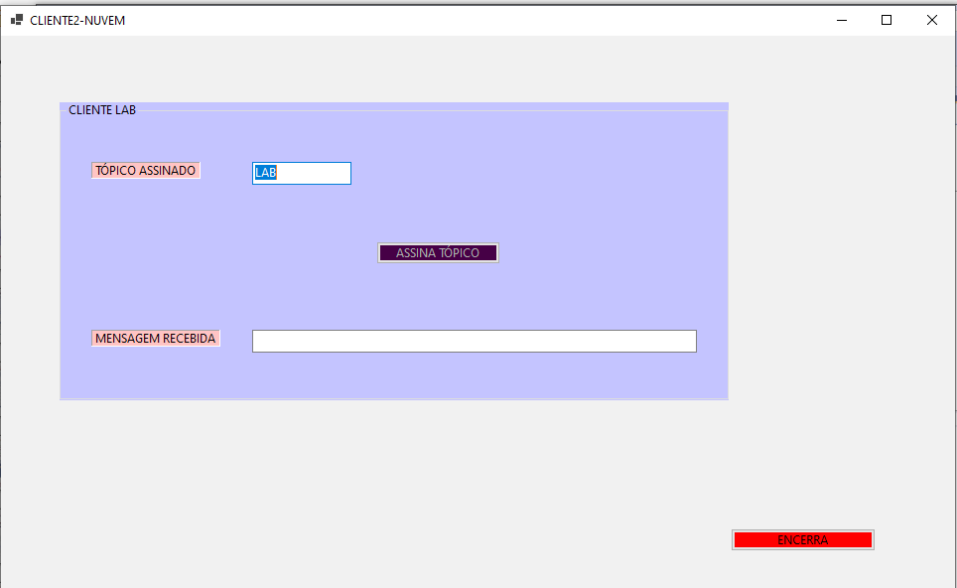


Com o formulário selecionado, mude sua propriedade **Text** para “CLIENTE2 NUVEM” ou um título do seu interesse

Clique na barra de ferramentas para visualizar os controles disponíveis

Insira os controles necessários ao projeto, tal como mostrado no slide a seguir...

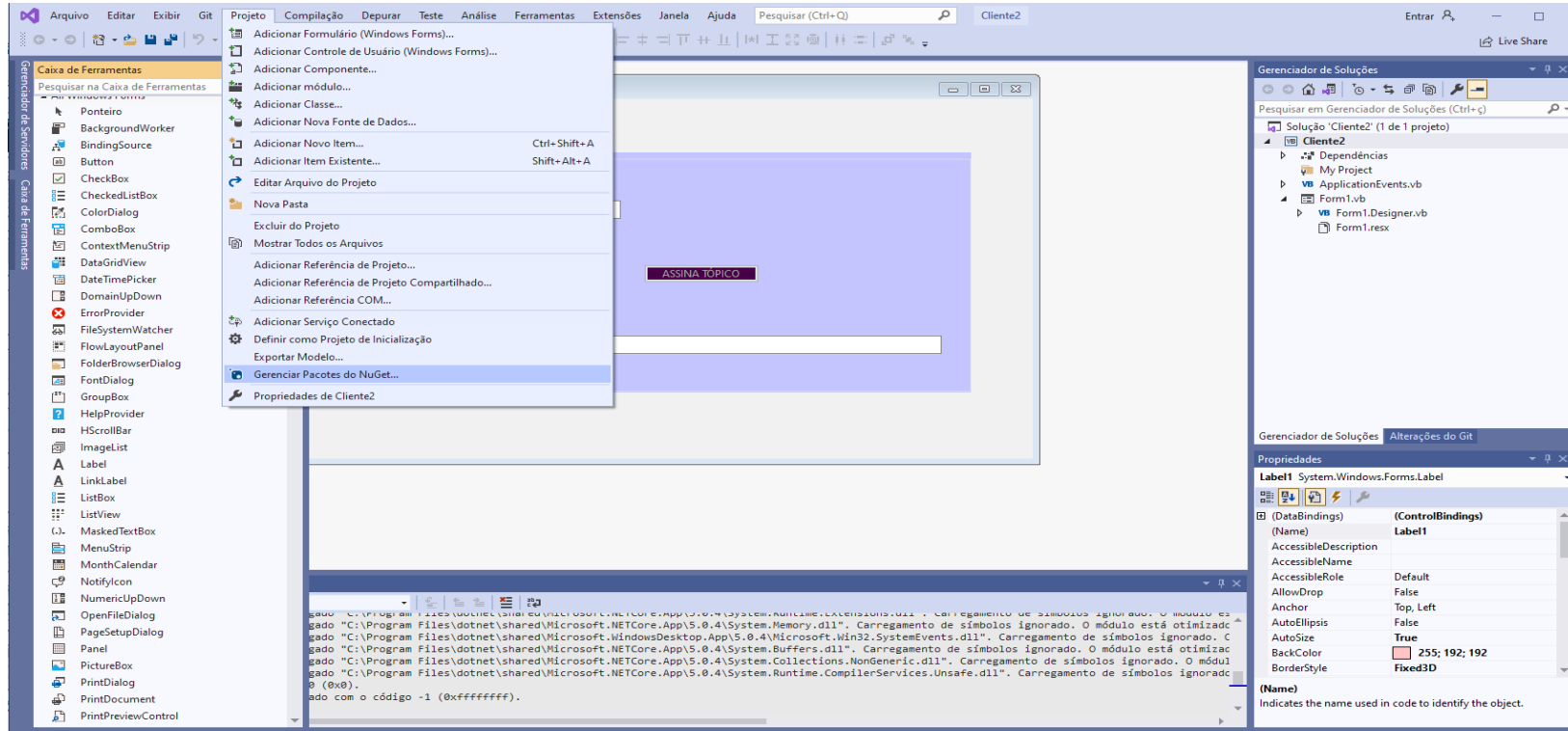
Leiaute do formulário



Após a inserção de todos os controles,
Altere as suas propriedades, tal como indicado na tabela

Controles	propriedades		
	Text	Back color	Name
Group Box	CLIENTE LAB	?	GroupBox1
Label 1	TÓPICO ASSINADO	?	Label1
Label 2	MENSAGEM RECEBIDA	?	Label1
Textbox1	LAB	?	Tb1
Textbox2		?	Tb2
Button	ASSINA TÓPICO	?	Btn1
Button	ENCERRA	?	Btn2

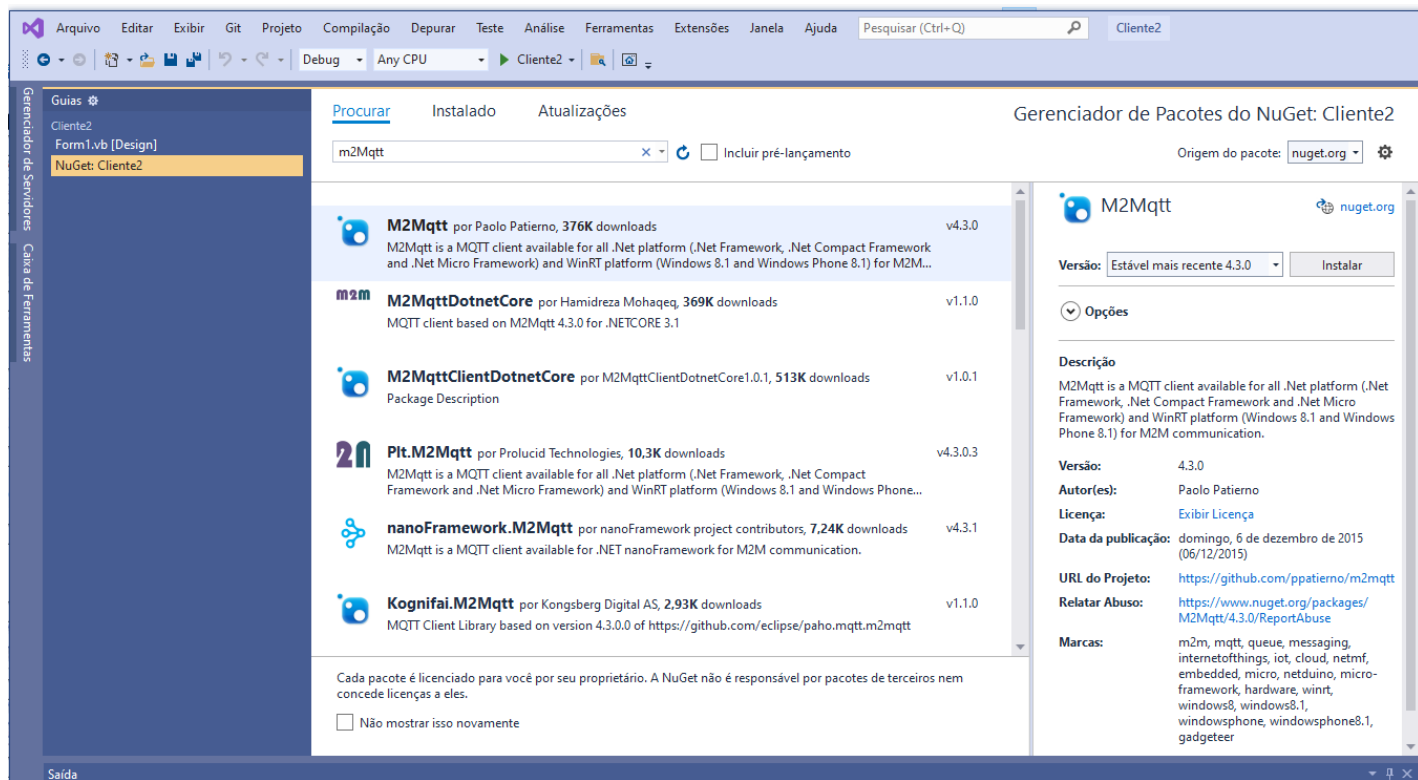
Inclusão de pacotes do NuGet em projetos MS-Visual Basic



Selecione o menu
“Projeto” e depois
“Gerenciar Pacotes
do Nuget...”

Pacotes contêm classes que podem ser utilizadas em outros aplicativos

Inclusão do pacote M2Mqtt desenvolvido por Paulo Patierno para estabelecer as conexões para publicação e assinatura no broker Mosquitto



Digite “M2Mqtt” no campo procurar
Para baixar o pacote desenvolvido
Por Paulo Patierno

Clique em “instalar”

Depois da instalação feche
A guia “Projeto” caso ela permaneça aberta

Dê um “duplo click” na área interna do formulário principal e inclua as classes utilizando o comando “imports”

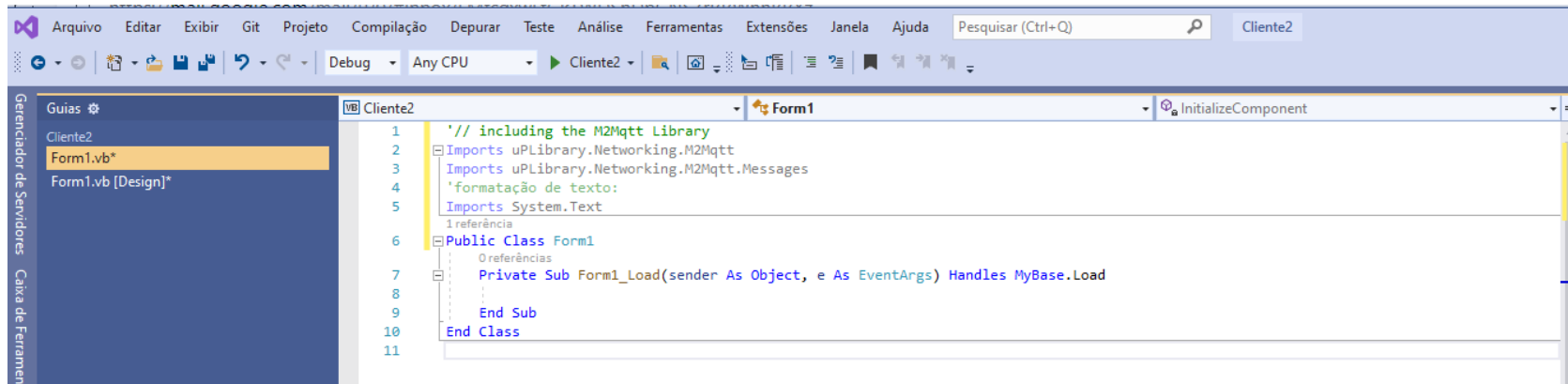
'classes do pacote M2Mqtt

Imports uPLibrary.Networking.M2Mqtt

Imports uPLibrary.Networking.M2Mqtt.Messages

'classe para tratamento de texto:

Imports System.Text



Estas classes devem ser incluídas antes do formulário principal (class Form1)

Declare as variáveis, conforme abaixo, antes do evento “Load”

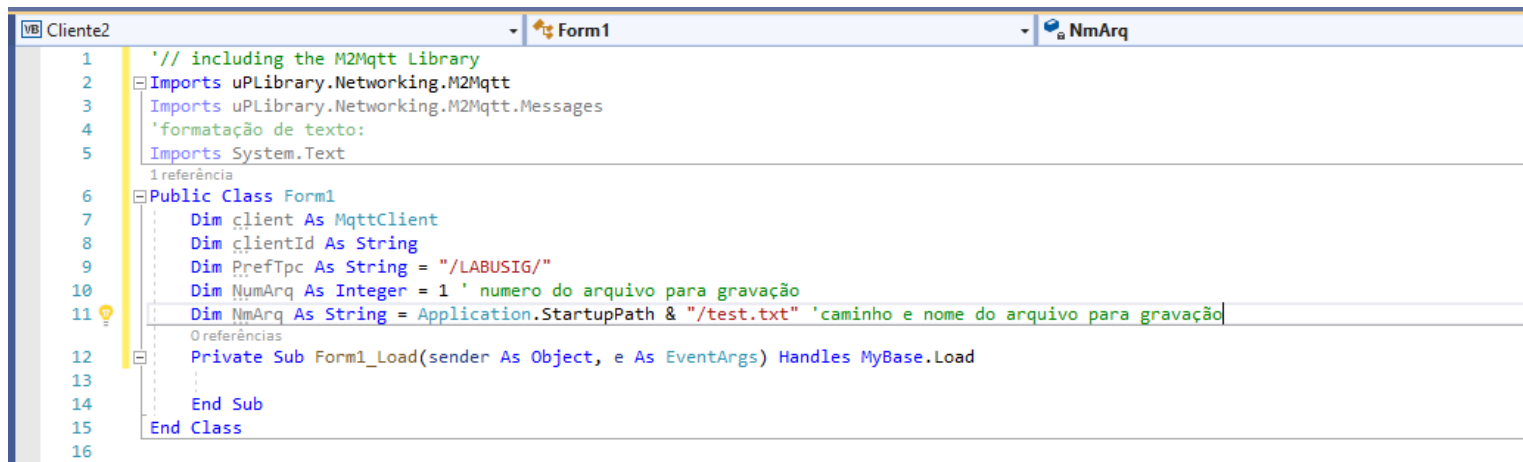
Dim client As MqttClient

Dim clientId As String

Dim PrefTpc As String = "/LABUSIG/"

Dim NumArq As Integer = 1 ' numero do arquivo para gravação

Dim NmArq As String = Application.StartupPath & "/test.txt" 'caminho e nome do arquivo para gravação



```
1  '/// including the M2Mqtt Library
2  Imports uPLibrary.Networking.M2Mqtt
3  Imports uPLibrary.Networking.M2Mqtt.Messages
4  'formatação de texto:
5  Imports System.Text
6  Public Class Form1
7      Dim client As MqttClient
8      Dim clientId As String
9      Dim PrefTpc As String = "/LABUSIG/"
10     Dim NumArq As Integer = 1 ' numero do arquivo para gravação
11     Dim NmArq As String = Application.StartupPath & "/test.txt" 'caminho e nome do arquivo para gravação
12     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
13
14     End Sub
15 End Class
16
```

O evento “Load” deverá ser programado conforme abaixo:

PROGRAMA DESENVOLVIDO POR PAULO PATIERNO

```
'Dim BrokerAddress As String = "85.119.83.194" ENDEREÇO AUXILIAR
```

```
Dim BrokerAddress As String = "test.mosquitto.org"
```

```
client = New MqttClient(BrokerAddress)
```

'cria uma gerenciamento do HANDLER ... PARA RECEBER AS MENSAGENS DISPARADAS

'UM EVENTO DO BROKER SERÁ PROCESSADO AUTOMATICAMENTE

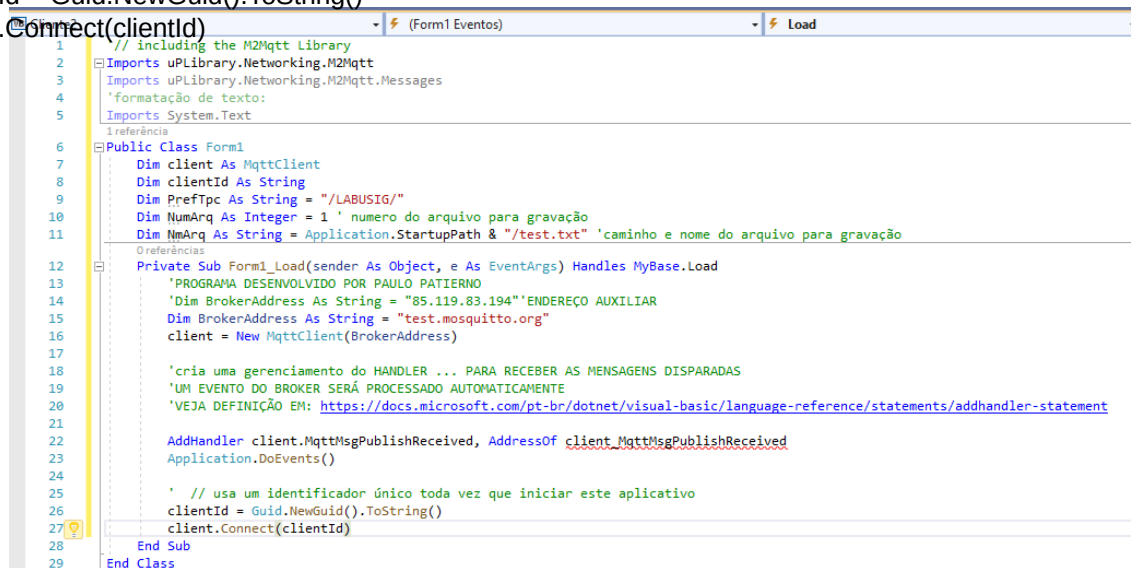
'VEJA DEFINIÇÃO EM: <https://docs.microsoft.com/pt-br/dotnet/visual-basic/language-reference/statements/addhandler-statement>

```
AddHandler client.MqttMsgPublishReceived, AddressOf client_MqttMsgPublishReceived  
Application.DoEvents()
```

' // usa um identificador único toda vez que iniciar este aplicativo

```
clientId = Guid.NewGuid().ToString()
```

```
client.Connect(clientId)
```



```
1 // including the M2Mqtt Library  
2 Imports uPLibrary.Networking.M2Mqtt  
3 Imports uPLibrary.Networking.M2Mqtt.Messages  
4 'formatação de texto:  
5 Imports System.Text  
6  
7 Public Class Form1  
8     Dim client As MqttClient  
9     Dim clientId As String  
10    Dim PrefTpc As String = "/LABUSIG/"  
11    Dim NumArq As Integer = 1 ' numero do arquivo para gravação  
12    Dim NmArq As String = Application.StartupPath & "/test.txt" 'caminho e nome do arquivo para gravação  
13  
14    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load  
15        'PROGRAMA DESENVOLVIDO POR PAULO PATIERNO  
16        'Dim BrokerAddress As String = "85.119.83.194" ENDEREÇO AUXILIAR  
17        Dim BrokerAddress As String = "test.mosquitto.org"  
18        client = New MqttClient(BrokerAddress)  
19  
20        'cria uma gerenciamento do HANDLER ... PARA RECEBER AS MENSAGENS DISPARADAS  
21        'UM EVENTO DO BROKER SERÁ PROCESSADO AUTOMATICAMENTE  
22        'VEJA DEFINIÇÃO EM: https://docs.microsoft.com/pt-br/dotnet/visual-basic/language-reference/statements/addhandler-statement  
23  
24        AddHandler client.MqttMsgPublishReceived, AddressOf client_MqttMsgPublishReceived  
25        Application.DoEvents()  
26  
27        ' // usa um identificador único toda vez que iniciar este aplicativo  
28        clientId = Guid.NewGuid().ToString()  
29        client.Connect(clientId)  
30    End Sub  
31 End Class
```

Crie uma rotina nova (Sub) abaixo do evento “Load” nomeando-a por “client_MqttMsgPublishReceived” e insira o código abaixo:

```
Sub client_MqttMsgPublishReceived(sender As Object, e As MqttMsgPublishEventArgs)
```

```
    Dim ReceivedMessage As String = Encoding.UTF8.GetString(e.Message)
```

```
    'tem que testar se existem linhas( threads) separadas
```

```
    'vide exemplo em:
```

```
    ' https://dotnetbrowser.support.teamdev.com/support/solutions/articles/9000119178--cross-thread-operation-not-valid-control-accessed-from-a-thread-other-than-the-thread-it-was-create
```

```
    If Me.Tb2.InvokeRequired Then
```

```
        BeginInvoke(Sub()
```

```
            'imprime para visualiação na tela:
```

```
            Me.Tb2.Text = ReceivedMessage
```

```
            'grava em arquivo:
```

```
            PrintLine(1, ReceivedMessage)
```

```
        End Sub)
```

```
    Else
```

```
        'imprime para visualiação na tela:
```

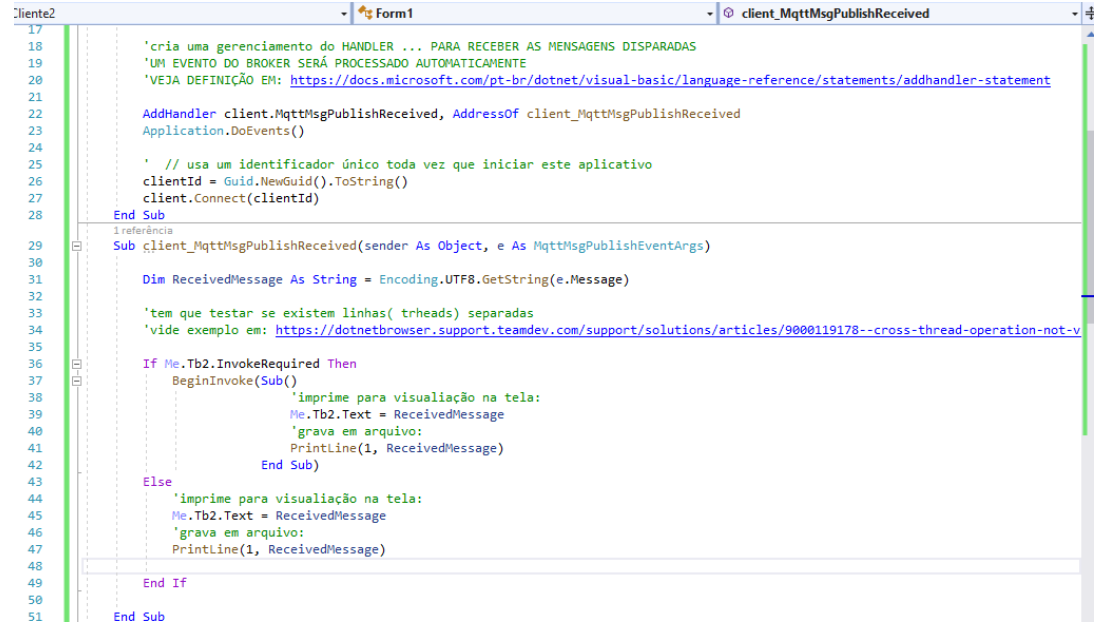
```
        Me.Tb2.Text = ReceivedMessage
```

```
        'grava em arquivo:
```

```
        PrintLine(1, ReceivedMessage)
```

```
    End If
```

```
End Sub
```



```
17 'cria uma gerenciamento do HANDLER ... PARA RECEBER AS MENSAGENS DISPARADAS
18 'UM EVENTO DO BROKER SERÁ PROCESSADO AUTOMATICAMENTE
19 'VEJA DEFINIÇÃO EM: https://docs.microsoft.com/pt-br/dotnet/visual-basic/language-reference/statements/addhandler-statement
20
21 AddHandler client.MqttMsgPublishReceived, AddressOf client_MqttMsgPublishReceived
22 Application.DoEvents()
23
24 ' // usa um identificador único toda vez que iniciar este aplicativo
25 clientId = Guid.NewGuid().ToString()
26 client.Connect(clientId)
27
28 End Sub
29
30 Sub client_MqttMsgPublishReceived(sender As Object, e As MqttMsgPublishEventArgs)
31
32     Dim ReceivedMessage As String = Encoding.UTF8.GetString(e.Message)
33
34     'tem que testar se existem linhas( threads) separadas
35     'vide exemplo em: https://dotnetbrowser.support.teamdev.com/support/solutions/articles/9000119178--cross-thread-operation-not-v
36
37     If Me.Tb2.InvokeRequired Then
38         BeginInvoke(Sub()
39             'imprime para visualiação na tela:
40             Me.Tb2.Text = ReceivedMessage
41             'grava em arquivo:
42             PrintLine(1, ReceivedMessage)
43         End Sub)
44     Else
45         'imprime para visualiação na tela:
46         Me.Tb2.Text = ReceivedMessage
47         'grava em arquivo:
48         PrintLine(1, ReceivedMessage)
49     End If
50
51 End Sub
```

Programa o evento “Click” do botão “ASSINA TÓPICO” conforme código abaixo

```
If (Me.Tb1.Text <> "") Then
    'abre arquivo para gravação
    FileOpen(NumArq, NmArq, OpenMode.Output)

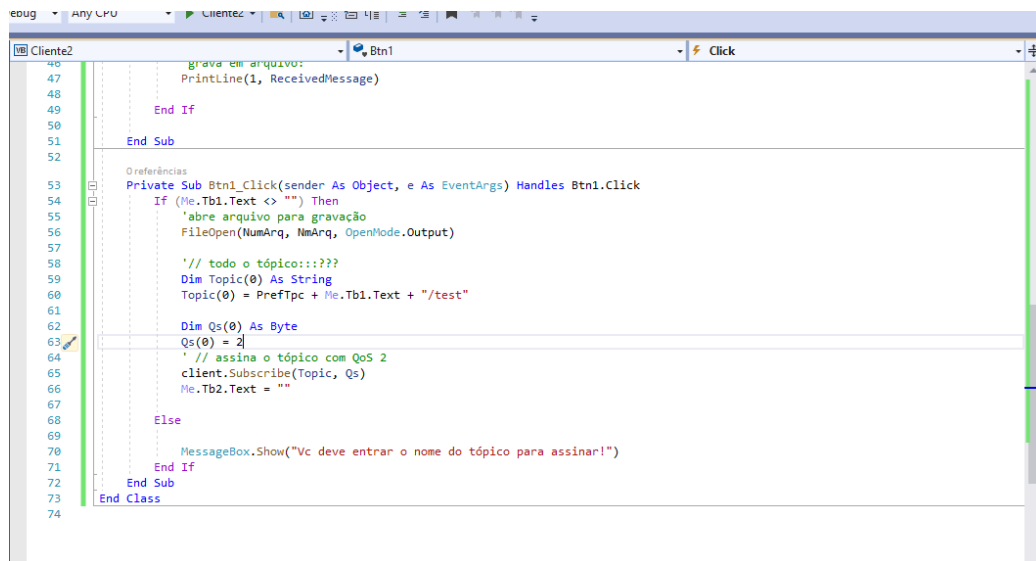
    '// todo o tópico:::???
    Dim Topic(0) As String
    Topic(0) = PrefTpc + Me.Tb1.Text + "/test"

    Dim Qs(0) As Byte
    Qs(0) = 2
    ' // assina o tópico com QoS 2
    client.Subscribe(Topic, Qs)
    Me.Tb2.Text = ""
```

Else

MessageBox.Show("Vc deve entrar o nome do tópico para assinar!")

End If

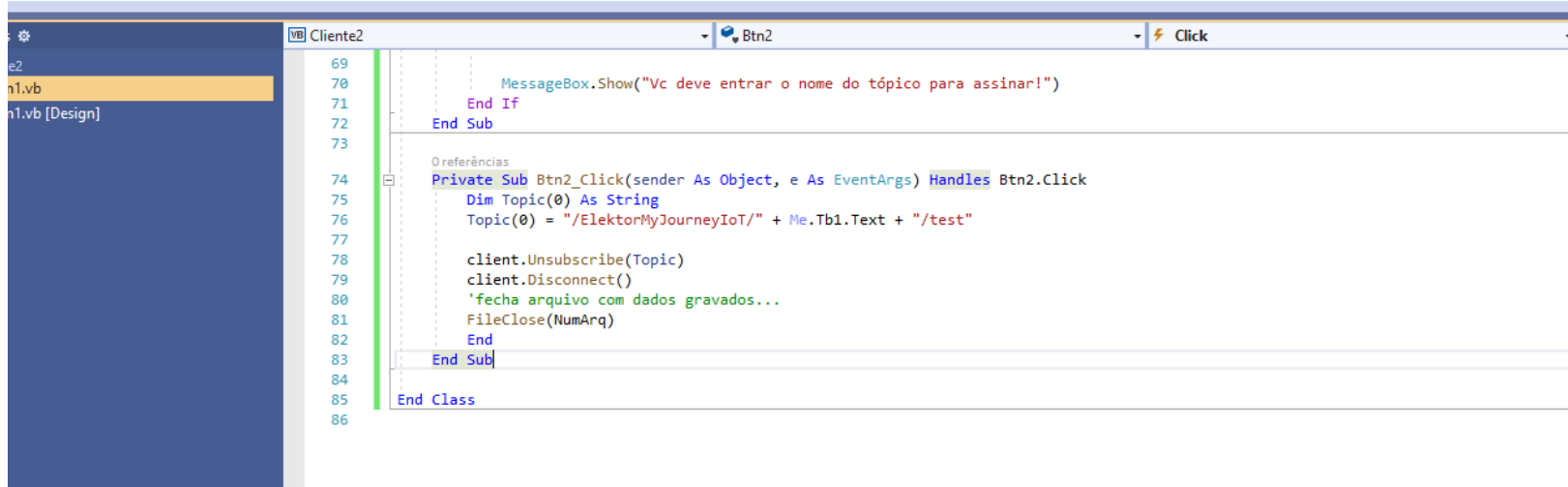


```
46 grava em arquivo:
47 PrintLine(1, ReceivedMessage)
48
49 End If
50
51 End Sub
52
53 Referências
54 Private Sub Btn1_Click(sender As Object, e As EventArgs) Handles Btn1.Click
55     If (Me.Tb1.Text <> "") Then
56         'abre arquivo para gravação
57         FileOpen(NumArq, NmArq, OpenMode.Output)
58
59         '// todo o tópico:::???
60         Dim Topic(0) As String
61         Topic(0) = PrefTpc + Me.Tb1.Text + "/test"
62
63         Dim Qs(0) As Byte
64         Qs(0) = 2
65         ' // assina o tópico com QoS 2
66         client.Subscribe(Topic, Qs)
67         Me.Tb2.Text = ""
68     Else
69         MessageBox.Show("Vc deve entrar o nome do tópico para assinar!")
70     End If
71 End Sub
72 End Class
73
74
```

Programe o evento “Click” do botão “ENCERRA” conforme código abaixo

```
Dim Topic(0) As String
Topic(0) = "/ElektorMyJourneyIoT/" + Me.Tb1.Text + "/test"

client.Unsubscribe(Topic)
client.Disconnect()
'fecha arquivo com dados gravados...
FileClose(NumArq)
End 'fecha aplicativo
```



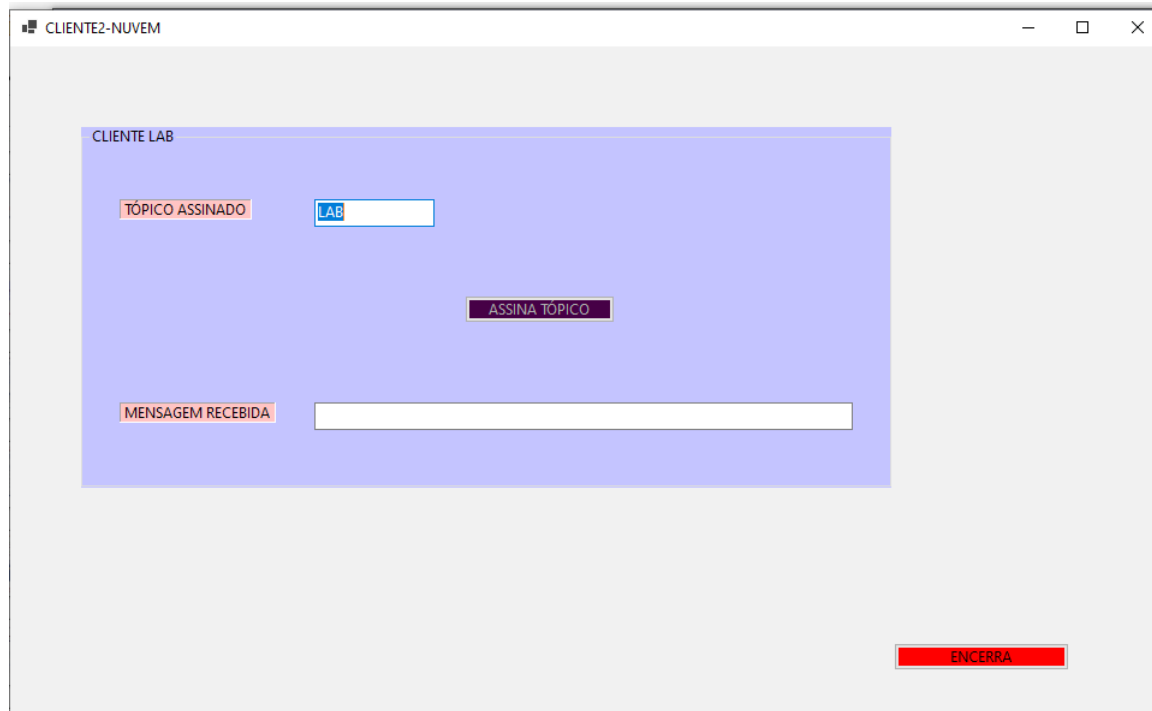
The screenshot shows the Visual Studio code editor with the following details:

- File Explorer (Left):** Shows a project structure with files like 'e2', 'n1.vb', and 'n1.vb [Design]'.
- Code Editor:** Displays the code for the 'Click' event of 'Btn2' in the 'Cliente2' class. The code is as follows:

```
69
70     MessageBox.Show("Vc deve entrar o nome do tópico para assinar!")
71 End If
72 End Sub
73
74 Private Sub Btn2_Click(sender As Object, e As EventArgs) Handles Btn2.Click
75     Dim Topic(0) As String
76     Topic(0) = "/ElektorMyJourneyIoT/" + Me.Tb1.Text + "/test"
77
78     client.Unsubscribe(Topic)
79     client.Disconnect()
80     'fecha arquivo com dados gravados...
81     FileClose(NumArq)
82 End
83 End Sub
84
85 End Class
86
```
- Toolbox (Right):** Shows the 'Click' event for the 'Btn2' control.

Compile o projeto para verificar inconsistências
“aperte a tecla F5”

A tela final deve ficar assim....



The screenshot shows a Windows application window titled "CLIENTE2-NUVEM". Inside the window, there is a light blue rectangular area representing the main interface. At the top left of this area is the text "CLIENTE LAB". Below it, on the left, is a label "TÓPICO ASSINADO" next to a small text input field containing the word "LAB". In the center of the blue area is a dark purple button labeled "ASSINA TÓPICO". At the bottom left of the blue area is a label "MENSAGEM RECEBIDA" next to a long, empty white text input field. In the bottom right corner of the main window, outside the blue area, is a red button labeled "ENCERRA".