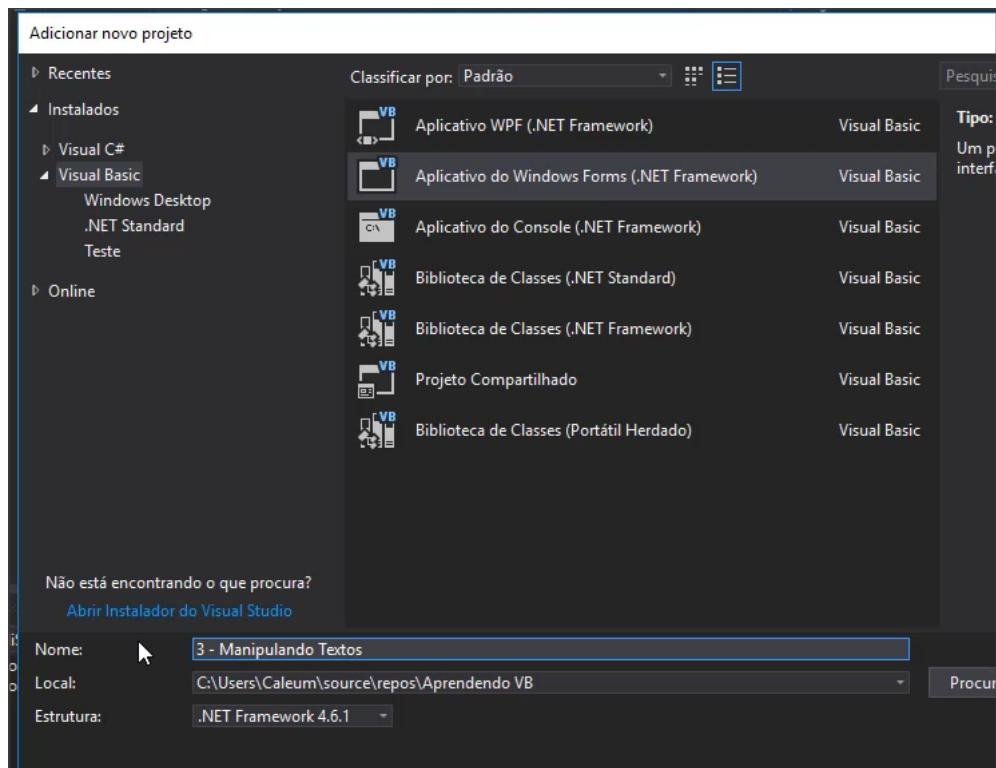


09

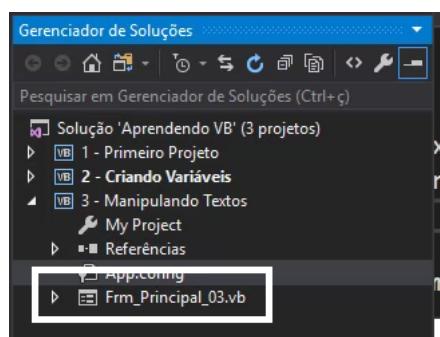
## Mão na massa

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

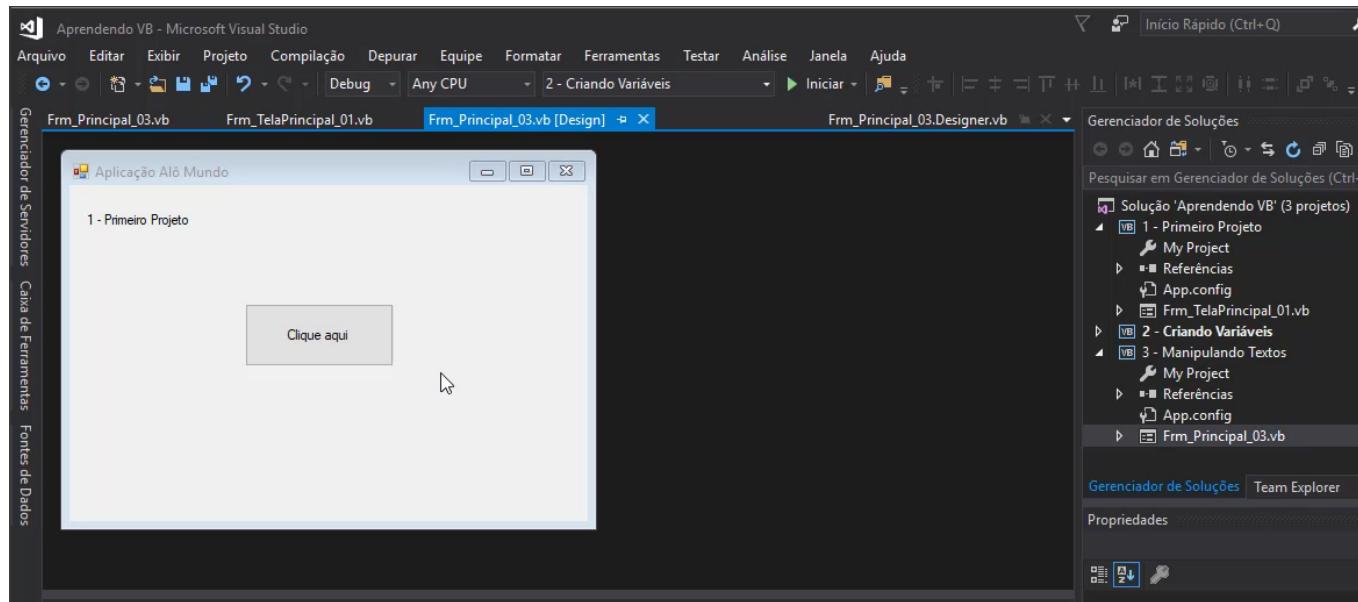
- 1) Crie um novo projeto na solução com o nome **3 - Manipulando Textos**. Não esqueça que este projeto é do tipo **Visual Basic -> Windows Desktop --> Aplicativo do Windows Forms**:



- 2) Renomeie o **Form1** para **Frm\_Principal\_03**:



- 3) Faça as cópias dos componentes do **projeto 1** para este projeto. Para isso proceda como já foi mostrado na aula passada (ou no exercício [Mão na massa](https://cursos.alura.com.br/course/vb-net-primeiros-passos/task/44047) (<https://cursos.alura.com.br/course/vb-net-primeiros-passos/task/44047>), passos 10 ao 19):



4) Altere a propriedade do componente dentro do código do Visual Basic. Altere também, usando o código do Visual Basic, o título da janela. Acesse o código fonte e, na sub-rotina `New`, que é executada toda vez que o formulário é criado, logo abaixo da função `InitializeComponent()`, que desenha os componentes na janela, adicione o seguinte código:

```
Public Class Frm_Principal_03

    Public Sub New()

        ' Esta chamada é requerida pelo designer.
        InitializeComponent()

        ' Adicione qualquer inicialização após a chamada InitializeComponent().

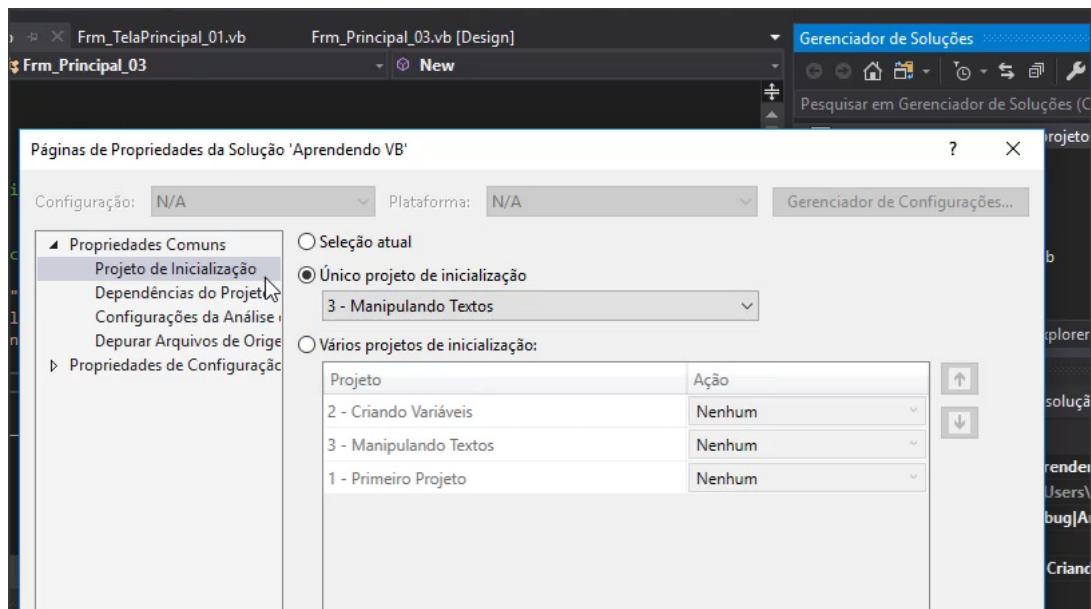
        Lbl_NomeProjeto.Text = "3 - Manipulando Textos"
        Btn_Principal.Text = "Clique para ver os resultados"
        Me.Text = "3 - Manipulando Textos"

    End Sub

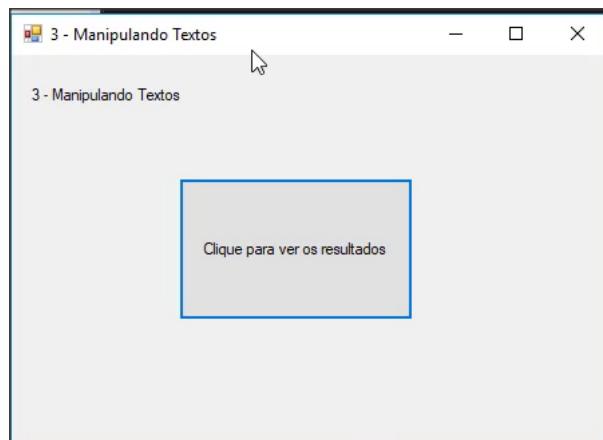
End Class
```

O `Text` do `Label` será `Lbl_NomeProjeto`, do botão será `Btn_Principal` e o título da janela será `3 - Manipulando Textos`.

5) Antes de executar esse exemplo, não esqueça de configurar o **projeto 3** como sendo o padrão para ser executado pela solução:



6) Ao executar a solução, tendo como padrão o **projeto 3**, você verá a seguinte janela:



7) Dê um duplo clique no botão do formulário deste projeto para inserir alguns códigos em Visual Basic. Declare uma variável que será um caractere, ou seja, ela será representada por uma letra, número ou símbolo. O tipo que representa um caractere no Visual Basic é **Char** :

```
Private Sub Btn_Principal_Click(sender As Object, e As EventArgs) Handles Btn_Principal.Click

    Dim PrimeiraLetra As Char = "a"
    MsgBox("O caractere da variável PrimeiraLetra é " + PrimeiraLetra)

End Sub
```

8) Cada caractere pode ser representado por um número inteiro, que você pode atribuir a uma variável **Char**, mas não pela sua representação, e sim pelo seu valor na tabela interna, a tabela ASCII. Com a função **Chr**, é possível associar o caractere pelo número interno:

```
Private Sub Btn_Principal_Click(sender As Object, e As EventArgs) Handles Btn_Principal.Click

    Dim PrimeiraLetra As Char = "a"
    MsgBox("O caractere da variável PrimeiraLetra é " + PrimeiraLetra)

    Dim SegundaLetra As Char = Chr(65)
    MsgBox("O caractere número 65 é " + SegundaLetra)
```

```
End Sub
```

O caractere 65 da tabela ASCII é o **A** maiúsculo.

9) Você pode repetir estes comandos de uma forma diferente. Você pode declarar uma variável do tipo inteiro e essa variável receberá um número, assim você pode usá-la como parâmetro para a função `Chr()`. Além disto, você pode efetuar cálculos matemáticos sobre essa variável inteira, para obter novos caracteres:

```
Private Sub Btn_Principal_Click(sender As Object, e As EventArgs) Handles Btn_Principal.Click
    Dim PrimeiraLetra As Char = "a"
    MsgBox("O caracter da variável PrimeiraLetra é " + PrimeiraLetra)

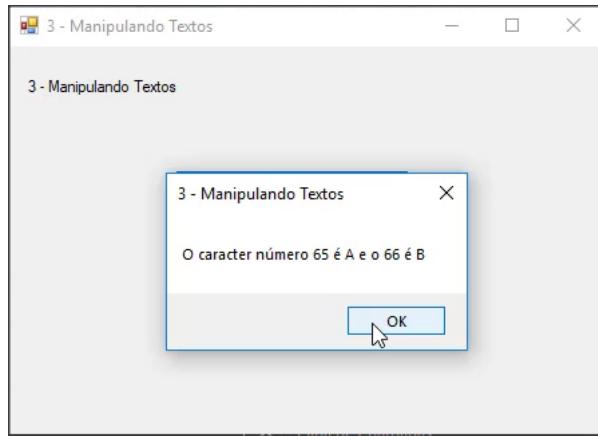
    Dim SegundaLetra As Char = Chr(65)
    MsgBox("O caracter número 65 é " + SegundaLetra)

    Dim NumeroChar As Integer
    NumeroChar = 65

    Dim TerceiraLetra As Char = Chr(NumeroChar)
    NumeroChar = NumeroChar + 1

    Dim QuartaLetra As Char = Chr(NumeroChar)
    MsgBox("O caracter número 65 é " + TerceiraLetra + " e o 66 é " + QuartaLetra)
End Sub
```

10) Executando aplicação, você verá o seguinte resultado:



11) Flexibilize um pouco mais a aplicação, adicionando dois componentes, um `TextBox` e um `Button`, com o `Name` `Txt_Numerico_Char` e `Btn_Char`, respectivamente.

12) Inicialize a propriedade `Text` do botão dinamicamente na construção da tela:

```
Public Sub New()
    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

```

```

Lbl_NomeProjeto.Text = "3 - Manipulando Textos"
Btn_Principal.Text = "Clique para ver os resultados"
Me.Text = "3 - Manipulando Textos"
Btn_Char.Text = "Clique para obter o char"

End Sub

```

13) Dê um duplo clique sobre o botão, e interprete o que será digitado na *TextBox*. Esse valor será interpretado internamente pelo Visual Basic como um texto, apesar de você saber que será digitado um número. Por isso, se você quiser fazer uma atribuição direta deste valor digitado a uma variável inteira, deve utilizar a função `Val()`, que converte o número, representado por uma string, em um número, seja ele inteiro ou ponto flutuante:

```

Private Sub Btn_Char_Click(sender As Object, e As EventArgs) Handles Btn_Char.Click

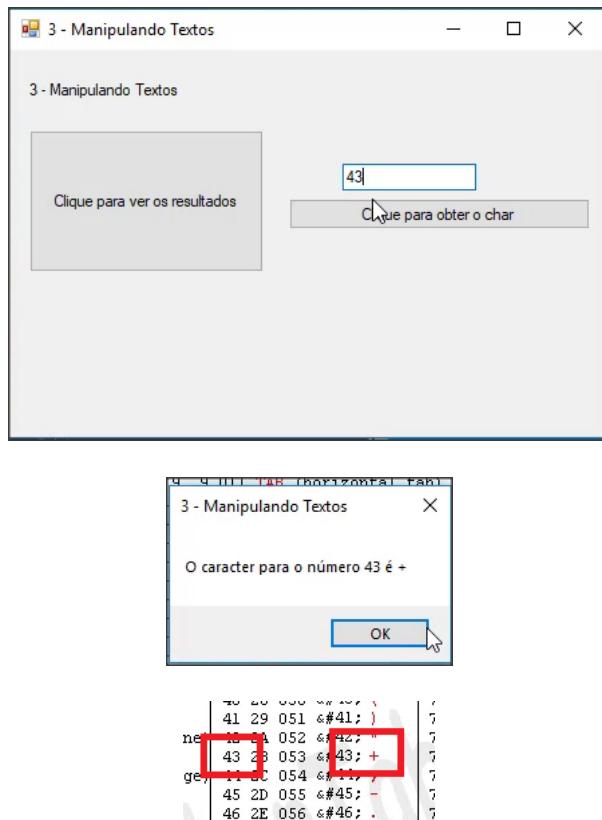
    Dim ValorChar As Integer = Val(Txt_Numero_Char.Text)
    Dim QuintaLetra As Char = Chr(ValorChar)

    MsgBox("O caracter para o número " + ValorChar.ToString + " é " + QuintaLetra)

End Sub

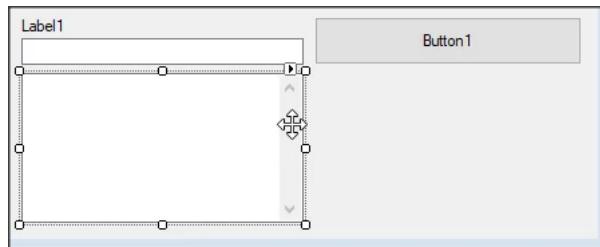
```

14) Executando a aplicação, você terá:



15) Volte para área de *Designer* do formulário do projeto 3 e insira um *Label*, com Name `Lbl_Curso`, um *TextBox*, com Name `Txt_Curso`, e um *Button*, com Name `Btn_Adicionar`. Adicione mais um *TextBox*, com duas novas propriedades, `Multiline` (para o *TextBox* ter múltiplas linhas) e `ScrollBars` com o valor `Vertical`, significa que, se as linhas que forem adicionadas ao *TextBox* forem maior do que o seu tamanho vertical, será automaticamente desenhado um *scroll* vertical.

16) O layout ficará assim:



17) Adicione mais um botão com Name Btn\_MostraLista :



18) Inicialize as propriedades Text dos Label e Button dinamicamente quando o formulário for aberto:

```
Public Sub New()

    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    Lbl_NomeProjeto.Text = "3 - Manipulando Textos"
    Btn_Principal.Text = "Clique para ver os resultados"
    Me.Text = "3 - Manipulando Textos"
    Btn_Char.Text = "Clique para obter o char"
    Lbl_Curso.Text = "Digite o seu curso preferido"
    Btn_Adicionar.Text = "Adicione a sua lista"
    Btn_MostraLista.Text = "Mostra a lista atual"

End Sub
```

19) No Txt\_Curso , será digitado o curso preferido do usuário. Depois, clicando no botão, este curso preferido será transferido para o Txt\_Lista . Programe essa lógica, dando um duplo clique sobre o botão Btn\_Adicionar :

```
Private Sub Btn_Adicionar_Click(sender As Object, e As EventArgs) Handles Btn_Adicionar.Click

    Dim CursoDigitado As String = Txt_Curso.Text

End Sub
```

20) Agora, é necessário adicionar este curso à lista. Esta lista será uma variável global acumulativa que irá receber todos os cursos escolhidos:

```
Public Class Frm_Principal_03

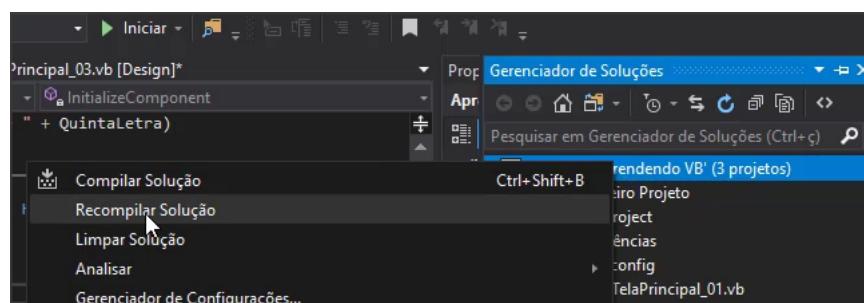
    Dim ListaCurso As String

    Public Sub New()
```

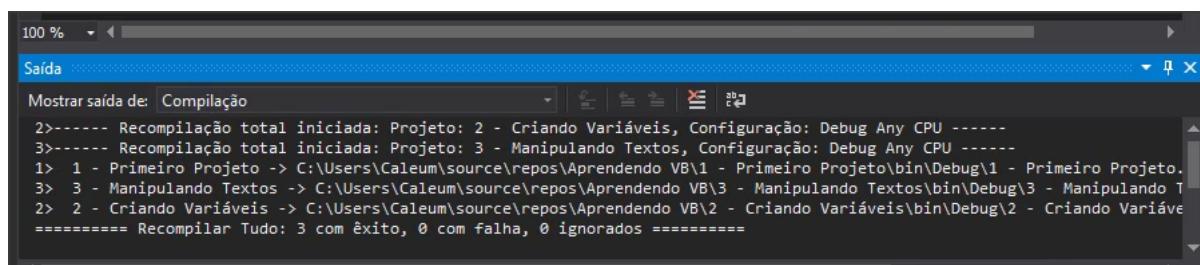
21) Voltando ao código do clique do botão, você pode adicionar a atribuição do curso digitado à variável `lista` e exibir o valor da variável `ListaCurso` no `TextBox` `Txt_Lista` :

```
Private Sub Btn_Adicionar_Click(sender As Object, e As EventArgs) Handles Btn_Adicionar.Click
    Dim CursoDigitado As String = Txt_Curso.Text
    ListaCurso = ListaCurso + CursoDigitado + vbCrLf
    Txt_Lista.Text = ListaCurso
End Sub
```

22) Veja agora como compilar a solução, para saber se há erros antes de executar. Para isso, basta clicar com o botão da direita do mouse sobre o nome da solução, no **Gerenciador de Soluções**, e escolher a opção **Recompilar Solução**:



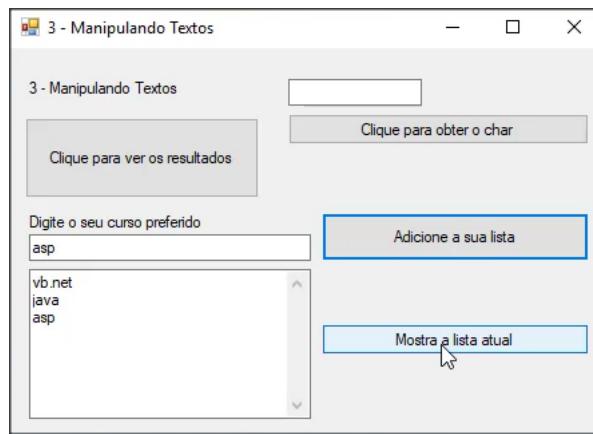
23) Você verá, na janela abaixo, chamada **Output**, o resultado da recompilação:



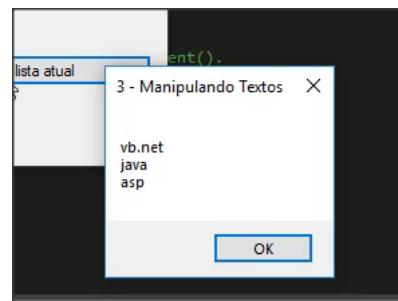
24) Dê um duplo clique sobre `Btn_MostraLista` , e inclua o código para mostrar o valor atual da variável `ListaCurso` :

```
Private Sub Btn_MostraLista_Click(sender As Object, e As EventArgs) Handles Btn_MostraLista.Click
    MsgBox(ListaCurso)
End Sub
```

25) Executando a aplicação, adicione alguns cursos à lista:



26) Clique no botão **Mostra a lista atual** e veja a caixa de mensagem com o valor da lista:



27) Volte ao *Form* principal do **projeto 3** e insira mais um botão com `Name = Btn_GuardarVariaveis`.

28) Acessando o código fonte, inicialize o texto deste botão:

```
Public Sub New()

    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    Lbl_NomeProjeto.Text = "3 - Manipulando Textos"
    Btn_Principal.Text = "Clique para ver os resultados"
    Me.Text = "3 - Manipulando Textos"
    Btn_Char.Text = "Clique para obter o char"
    Lbl_Curso.Text = "Digite o seu curso preferido"
    Btn_Adicionar.Text = "Adicione a sua lista"
    Btn_MostrarLista.Text = "Mostra a lista atual"
    Btn_GuardarVariaveis.Text = "Clique aqui"

End Sub
```

29) E no código do clique deste novo botão, insira as seguintes linhas:

```
Private Sub Btn_GuardarVariaveis_Click(sender As Object, e As EventArgs) Handles Btn_GuardarVariaveis.Click

    Dim idade As Integer = 32
    Dim idadeGustavo As Integer

    idadeGustavo = idade
```

```

MsgBox("Idade é igual a " + idade.ToString + " enquanto a idadeGustavo é " + idadeGustavo.ToString)

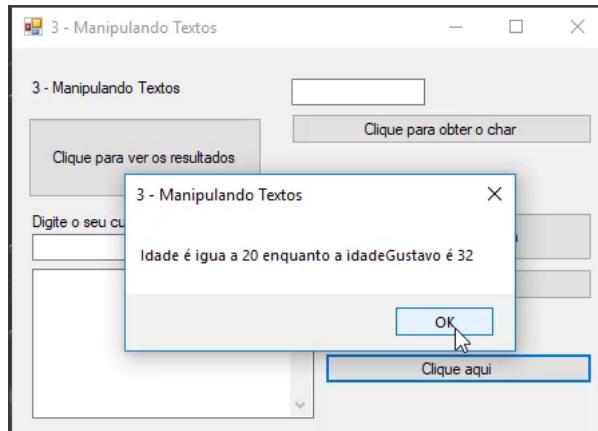
idade = 20
MsgBox("Idade é igual a " + idade.ToString + " enquanto a idadeGustavo é " + idadeGustavo.ToString)

End Sub

```

Como `idadeGustavo` é igual à `idade`, pode-se esperar que, ao atribuir o valor 20 para a `idade`, `idadeGustavo` também será igual a 20.

30) Veja o resultado:



Isso significa que, ao atribuir o valor da variável `idade` à variável `idadeGustavo`, as duas deixam de ter um vínculo e o valor da variável `idadeGustavo` passa a ser independente.

31) O código final ficará assim:

```

Public Class Frm_Principal_03

    Dim ListaCurso As String

    Public Sub New()

        ' Esta chamada é requerida pelo designer.
        InitializeComponent()

        ' Adicione qualquer inicialização após a chamada InitializeComponent().

        Lbl_NomeProjeto.Text = "3 - Manipulando Textos"
        Btn_Principal.Text = "Clique para ver os resultados"
        Me.Text = "3 - Manipulando Textos"
        Btn_Char.Text = "Clique para obter o char"
        Lbl_Curso.Text = "Digite o seu curso preferido"
        Btn_Adicionar.Text = "Adicione a sua lista"
        Btn_MostraLista.Text = "Mostra a lista atual"
        Btn_GuardarVariaveis.Text = "Clique aqui"

    End Sub

    Private Sub Btn_Principal_Click(sender As Object, e As EventArgs) Handles Btn_Principal.Click

        Dim PrimeiraLetra As Char = "a"

```

```
MsgBox("O caracter da variável PrimeiraLetra é " + PrimeiraLetra)

Dim SegundaLetra As Char = Chr(65)
MsgBox("O caracter número 65 é " + SegundaLetra)

Dim NumeroChar As Integer
NumeroChar = 65
Dim TerceiraLetra As Char = Chr(NumeroChar)

NumeroChar = NumeroChar + 1
Dim QuartaLetra As Char = Chr(NumeroChar)

MsgBox("O caracter número 65 é " + TerceiraLetra + " e o 66 é " + QuartaLetra)

End Sub

Private Sub Btn_Char_Click(sender As Object, e As EventArgs) Handles Btn_Char.Click

    Dim ValorChar As Integer = Val(Txt_Numeros_Char.Text)
    Dim QuintaLetra As Char = Chr(ValorChar)

    MsgBox("O caracter para o número " + ValorChar.ToString + " é " + QuintaLetra)

End Sub

Private Sub Btn_Adicionar_Click(sender As Object, e As EventArgs) Handles Btn_Adicionar.Click

    Dim CursoDigitado As String = Txt_Curso.Text
    ListaCurso = ListaCurso + CursoDigitado + vbCrLf
    Txt_Lista.Text = ListaCurso

End Sub

Private Sub Btn_MostraLista_Click(sender As Object, e As EventArgs) Handles Btn_MostraLista.Click

    MsgBox(ListaCurso)

End Sub

Private Sub Btn_GuardarVariaveis_Click(sender As Object, e As EventArgs) Handles Btn_GuardarVariaveis.Click

    Dim idade As Integer = 32
    Dim idadeGustavo As Integer

    idadeGustavo = idade

    MsgBox("Idade é igual a " + idade.ToString + " enquanto a idadeGustavo é " + idadeGustavo.ToString)

    idade = 20

    MsgBox("Idade é igual a " + idade.ToString + " enquanto a idadeGustavo é " + idadeGustavo.ToString)

End Sub

End Class
```

