

## Mãos na massa

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

---

1) Na solução **ByteBank**, crie um novo projeto do tipo **Visual Basic --> Windows Desktop --> Aplicativo do Windows Forms**, chamado **ExemploInterface**.

2) Ao novo projeto, adicione um novo item do tipo **Interface**, de nome **IAbrirPorta** e inclua o código abaixo:

```
Public Interface IAbrirPorta

    Function AbrirPorta() As String

End Interface
```

3) Crie uma classe chamada **Casa**, com o seguinte código:

```
Public Class Casa
    Implements IAbrirPorta

    Public Function AbrirPorta() As String Implements IAbrirPorta.AbrirPorta
        Return "Estou abrindo a porta de uma CASA."
    End Function

End Class
```

4) Crie outra classe chamada **Predio**, com o seguinte código:

```
Public Class Predio
    Implements IAbrirPorta

    Public Function AbrirPorta() As String Implements IAbrirPorta.AbrirPorta
        Return "Estou abrindo a porta de um PREDIO."
    End Function

End Class
```

5) Crie mais uma classe, chamada **Sobrado**, com o seguinte código:

```
Public Class Sobrado
    Implements IAbrirPorta

    Public Function AbrirPorta() As String Implements IAbrirPorta.AbrirPorta
        Return "Estou abrindo a porta de um SOBRADO."
    End Function

End Class
```

6) Crie a última classe, chamada `Imovel`, com o seguinte código:

```
Public Class Imovel

    Public Function AbrirPorta(Imovel As IAbrirPorta) As String
        Return Imovel.AbrirPorta
    End Function

End Class
```

7) Crie um novo formulário, chamado `ExemploInterface`, com `Name` `ExemploInterface`, e com as propriedades `StartPosition` e `WindowState` com os valores `CenterScreen` e `Normal`, respectivamente.

8) No novo formulário, adicione um `Button`, de `Name` `Btn_ExemploInterface`.

9) No código-fonte do formulário `ExemploInterface`, na sub-rotina `New()`, inicialize as propriedades `Text` dos componentes:

```
Public Sub New()

    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    Me.Text = "Exemplo Interface"
    Btn_ExemploInterface.Text = "Clique Aqui"

End Sub
```

10) Na rotina `Click` de `Btn_ExemploInterface`, instancie um `Imovel` e exiba as mensagens ao abrir uma porta de uma `Casa`, um `Predio` e um `Sobrado`:

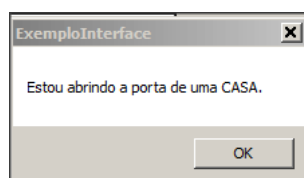
```
Private Sub Btn_ExemploInterface_Click(sender As Object, e As EventArgs) Handles Btn_ExemploInterfac

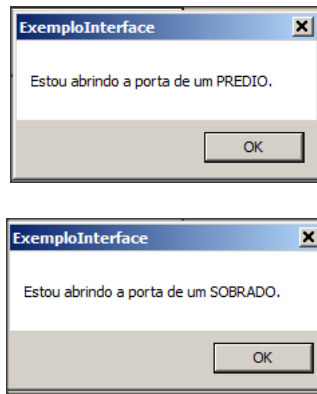
    Dim vX As New Imovel

    MsgBox(vX.AbrirPorta(New Casa))
    MsgBox(vX.AbrirPorta(New Predio))
    MsgBox(vX.AbrirPorta(New Sobrado))

End Sub
```

11) Faça com que o novo projeto seja o projeto padrão da solução. Após isso, salve-o, execute-o e clique no botão **Clique Aqui**:





12) Volte ao Visual Studio e agora coloque o projeto **AutenticacaoByteBank** como projeto padrão da solução.

13) No diretório **Bibliotecas**, crie uma interface chamada **IAutentica**, com o seguinte código:

```
Public Interface IAutentica
    Function Autenticar(senhaTentativa As String) As Boolean
End Interface
```

14) Dentro de **Classes**, crie um outro diretório chamado **Externos** e dentro dele, crie a classe **Representante**, com o seguinte código:

```
Namespace Classes.Externos

    Public Class Representante
        Implements IAutentica

        #Region "PROPRIEDADES"
            Public Property senha As String
        #End Region

        #Region "MÉTODOS"

            Public Function Autenticar(senhaTentativa As String) As Boolean Implements IAutentica.Autent
                If senhaTentativa.Length < 10 Then
                    Return False
                End If
                If senha = senhaTentativa Then
                    Return True
                End If
                Return False
            End Function

        #End Region

    End Class

End Namespace
```

15) Na classe **FuncionarioAutenticavel**, inclua a interface na declaração da classe:

```
Public MustInherit Class FuncionarioAutenticavel
    Inherits Funcionario
    Implements IAutentica
```

16) Modifique o método `Autenticar` para o código abaixo:

```
Public Function Autenticar(senhaTentativa As String) As Boolean Implements IAutentica.Autenticar
    If senhaTentativa.Length < 5 Then
        Return False
    End If
    If senha = senhaTentativa Then
        Return True
    End If
    Return False
End Function
```

17) No formulário `Frm_Principal`, adicione um novo botão, com `Name` `Btn_Autentica2`:

18) Na sub-rotina `New()` do formulário, inicialize a propriedade `Text` do novo botão:

```
Btn_Autentica2.Text = "Autentica com Representante"
```

19) Na rotina `Click` de `Btn_Autentica2`, crie um diretor e um representante, atribua uma senha a eles, tente logar no sistema interno e imprima o resultado::

```
Private Sub Btn_Autentica2_Click(sender As Object, e As EventArgs) Handles Btn_Autentica2.Click

    Dim Pedro As New Diretor("123.123.123-00", 2000)
    Pedro.senha = "abcefgjh"

    Dim Alberto As New Representante()
    Alberto.senha = "abcefgjhijklmnopq"

    Dim vLoga As Boolean
    Dim SistemaInterno As New SistemaInterno

    vLoga = SistemaInterno.Logar(Pedro, "abcefgjh")
    MsgBox("A tentativa de logar no sistema (Funcionario) resultou em " + vLoga.ToString)

    vLoga = SistemaInterno.Logar(Alberto, "abcefgjhijklmnopq")
    MsgBox("A tentativa de logar no sistema (Representante) resultou em " + vLoga.ToString)

End Sub
```

20) Salve e execute o projeto. Clique no botão **Autenticação com Representante**:

