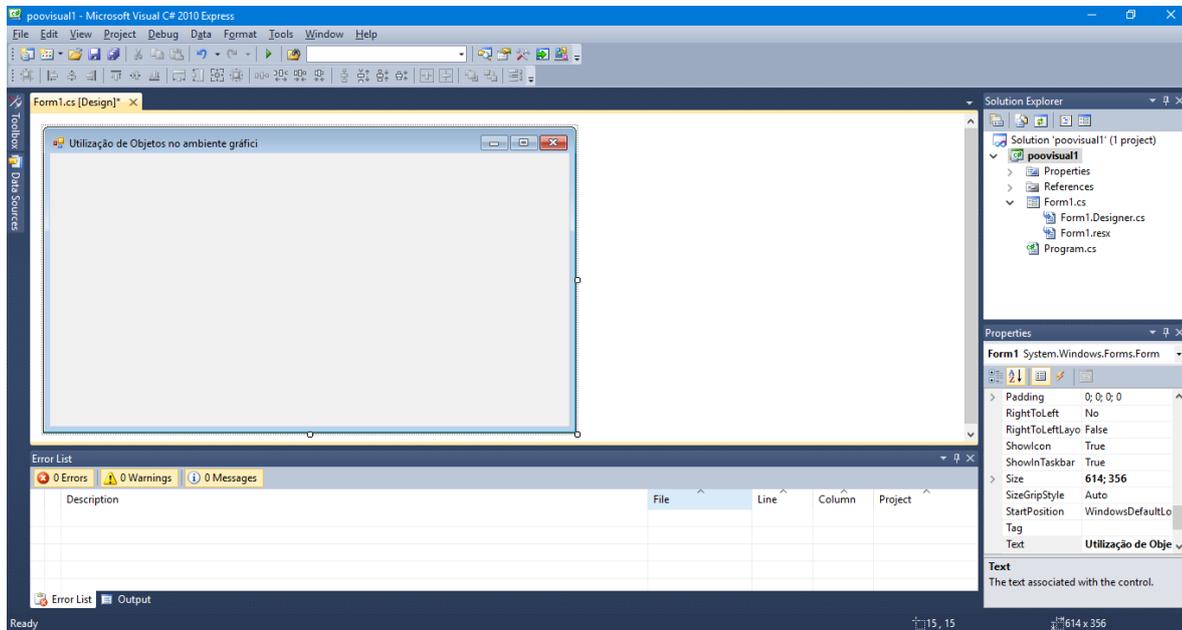


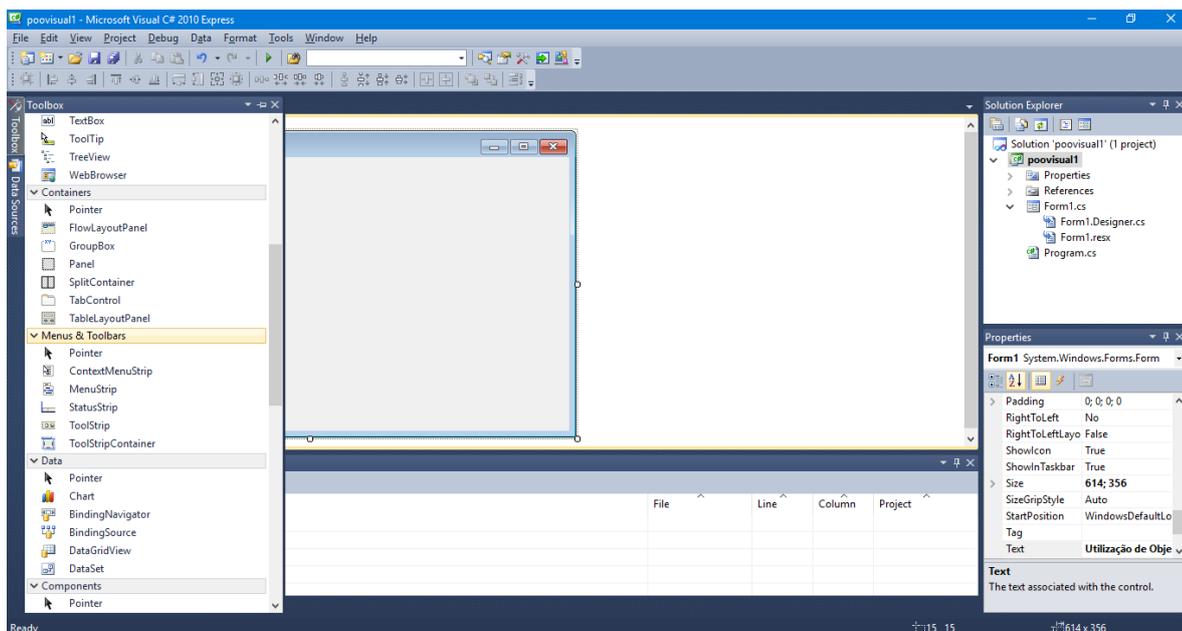
## 05 – Como utilizar POO no ambiente gráfico do C#?

Abra um novo projeto, mas desta vez, não escolha uma aplicação Console. Escolha Windows Form Application.

A diferença para a aplicação de Console é a presença de um objeto Form (Formulário), que é uma janela padrão Windows . Neste formulário, colocaremos os objetos visuais para compor nossa aplicação.



Para inserir objetos visuais, utilize a barra de ferramentas (toolbox) que deve estar à esquerda, oculta, mas que ao aproximar o mouse ela deve se abrir (instalação padrão do VS. Estas configurações podem ser alteradas).

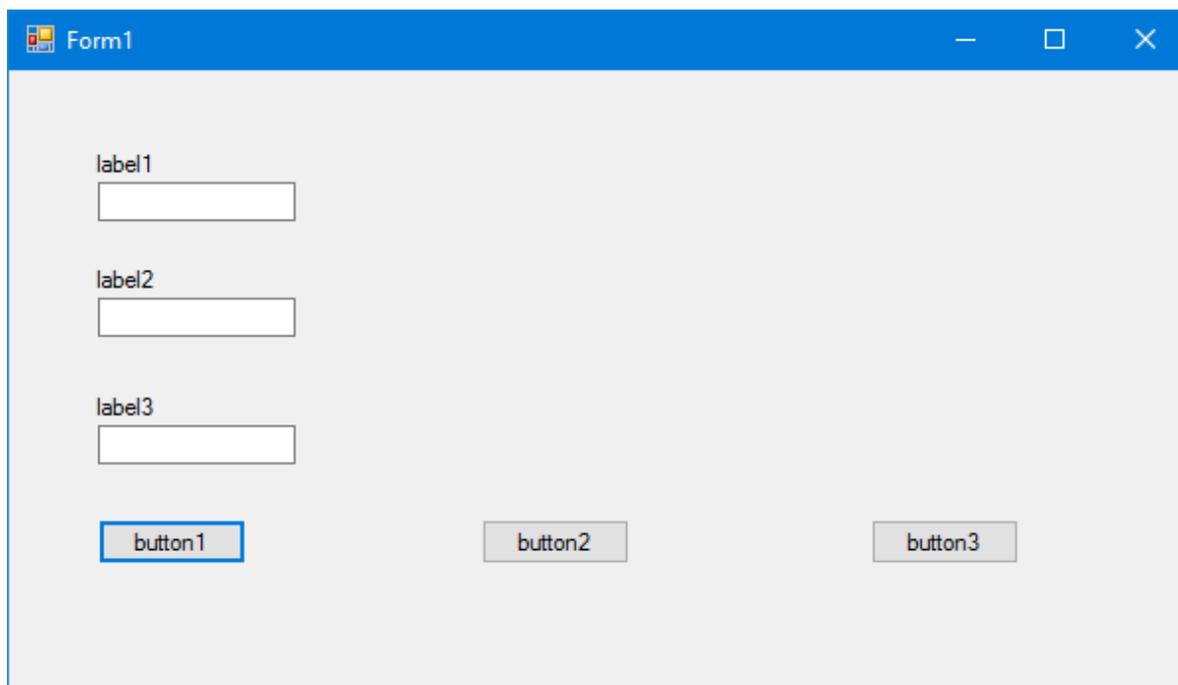


Em seguida, escolha o objeto desejado, clique e arraste até o formulário. Pode-se também dar um duplo clique do mouse sobre o objeto.

Ao selecionar qualquer objeto (desde o formulário até os objetos inseridos), podemos alterar seus atributos na janela de Propriedades (geralmente na direita, embaixo, ou pressionando a tecla F4 do teclado, lembrando que também podemos ocultar ou mover de posição esta janela).

Existem alguns atributos padrões a todos os objetos e existem atributos particulares para cada objeto. Por exemplo, a maioria dos objetos possuem a propriedade Text. Ela controla o texto que é exibido no objeto. A propriedade (Name), que aparece entre parênteses mesmo, define o nome do objeto, como se fosse um nome de variável.

Vamos inserir então 9 objetos, sendo 3 label, 3 textbox e 3 button, distribuindo-os como no exemplo:



Em seguida, clicando nos objetos e alterando as propriedades Text, coloque os textos como na figura em seguida:

A screenshot of a Windows application window titled "Form1". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is light gray and contains three text input fields stacked vertically. The first is labeled "Id:" and is a small rectangular box. The second is labeled "Nome:" and is a longer rectangular box. The third is labeled "E-mail:" and is also a longer rectangular box. Below the text boxes are three buttons arranged horizontally. The first button is labeled "Gravar no objeto" and has a blue border. The second button is labeled "Limpar dados" and the third is labeled "Recuperar do objeto". Both the second and third buttons are disabled, appearing in a lighter gray color.

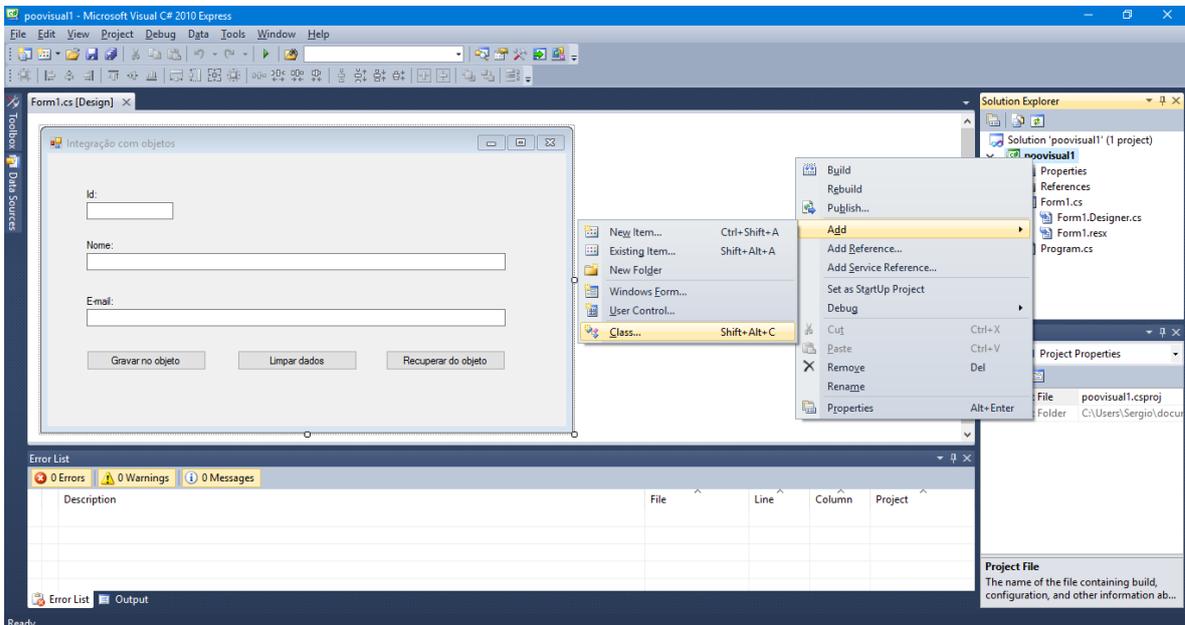
Será preciso adequar o tamanho de alguns objetos. Podemos utilizar a propriedade Size de cada um deles, ou clicando com o mouse e redimensionando no próprio formulário.

Vamos alterar as propriedades (Name) dos objetos TextBox e Button. Na sequência, coloque txtId, txtNome e txtEmail nas textboxes e btnGravar, btnLimpar e btnRecuperar nos button.

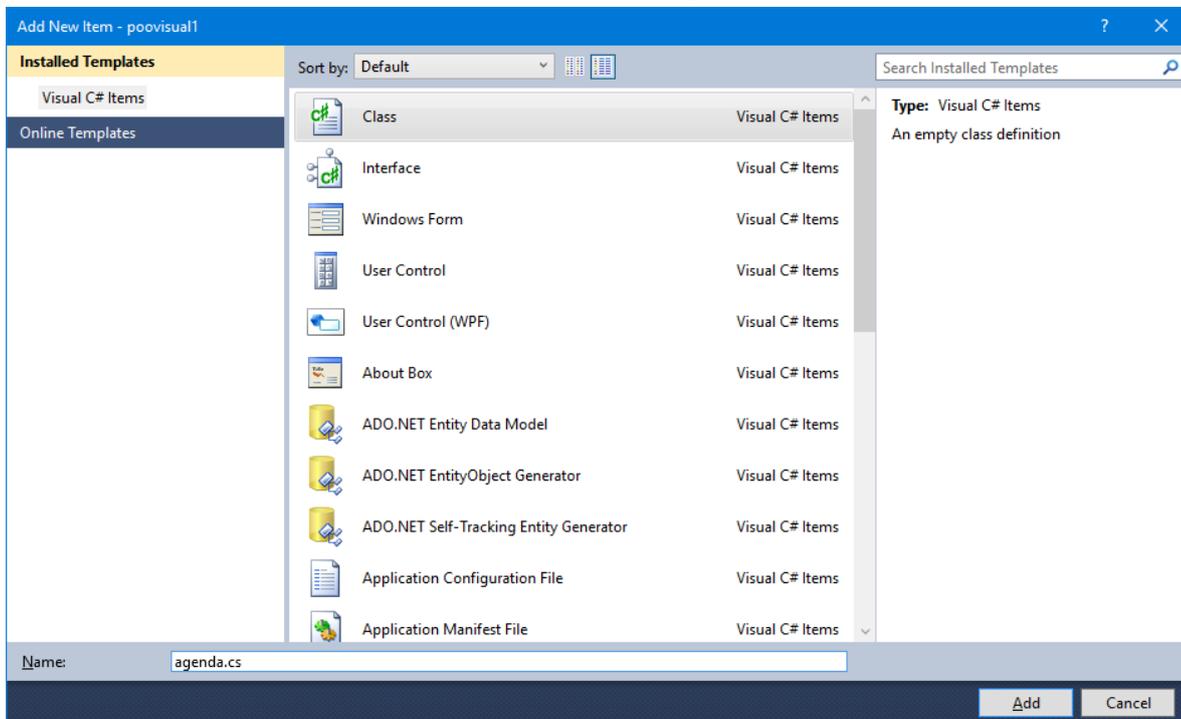
O título do formulário também pode ser alterado na propriedade Text do formulário.

Em seguida, vamos começar a programação. Para criar uma nova classe (como criamos nas aplicações Console), não poderemos utilizar o arquivo que define o código gráfico. Iremos criar um novo arquivo no projeto específico para isso. Para isso utilize uma das opções:

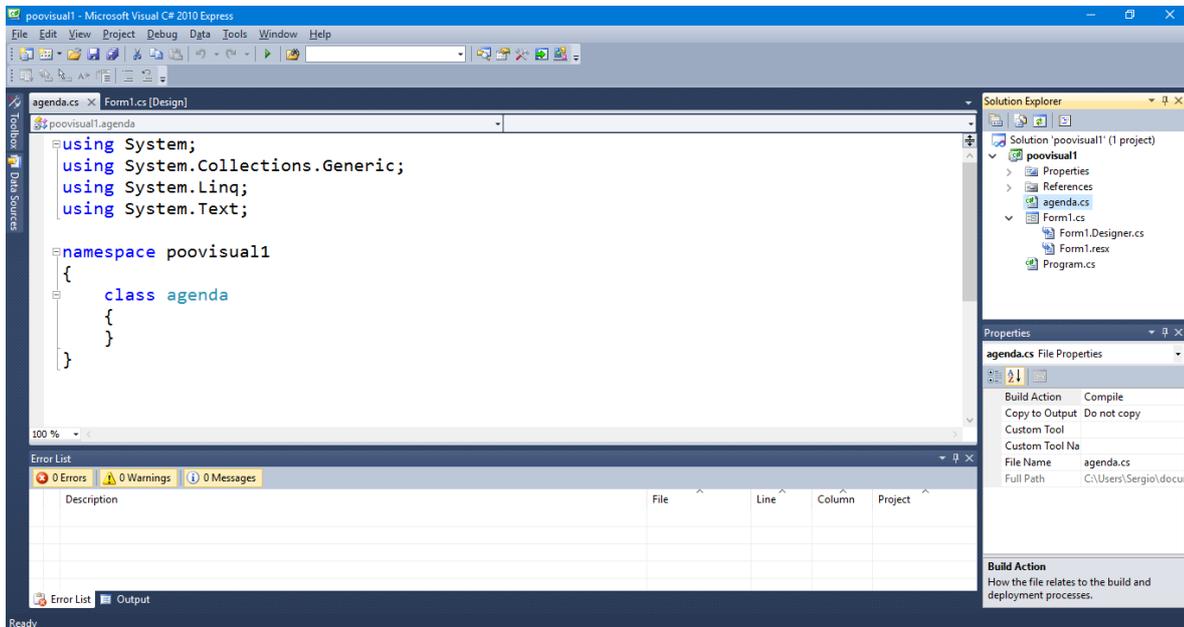
- 1) Menu Project, ADD Class
- 2) Atalho do teclado Shift + Alt + C
- 3) Em Solution Explorer (sobre a janela de propriedades na direita), botão direito do mouse, add Class – exemplo na figura a seguir.



Dê o nome para sua classe (neste caso, chamei de Agenda)



E sua classe estará criada e pronto para ser implementada:



Repare que foi criada uma aba na janela principal, e para voltar ao formulário basta selecionar a aba Form1.CS. Na Solution Explorer também é possível dar dois cliques no objeto que deseja visualizar na janela principal (seja na classe ou no formulário).

Vamos criar a classe agenda, criando os atributos e os métodos a seguir:

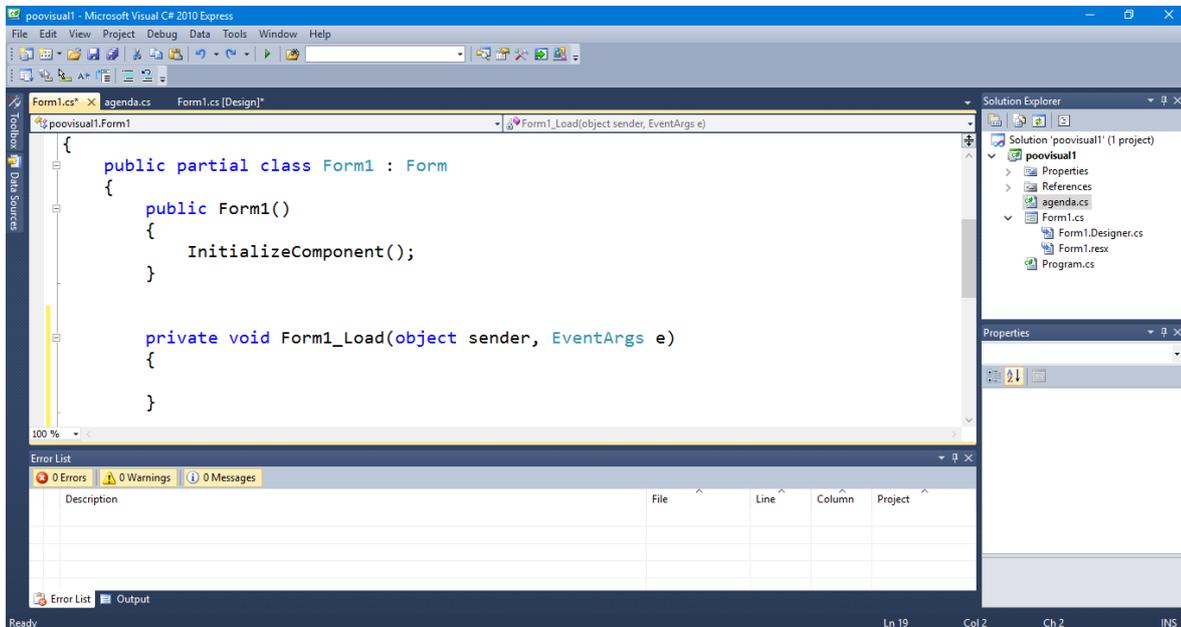
```
class agenda
{
    public int Id { get; private set; }
    public string Nome { get; private set; }
    public string Email { get; private set; }

    public void GravaPessoa(int NovoId, string NovoNome, string NovoEmail)
    {
        this.Id = NovoId;
        this.Nome = NovoNome;
        this.Email = NovoEmail;
    }
}
```

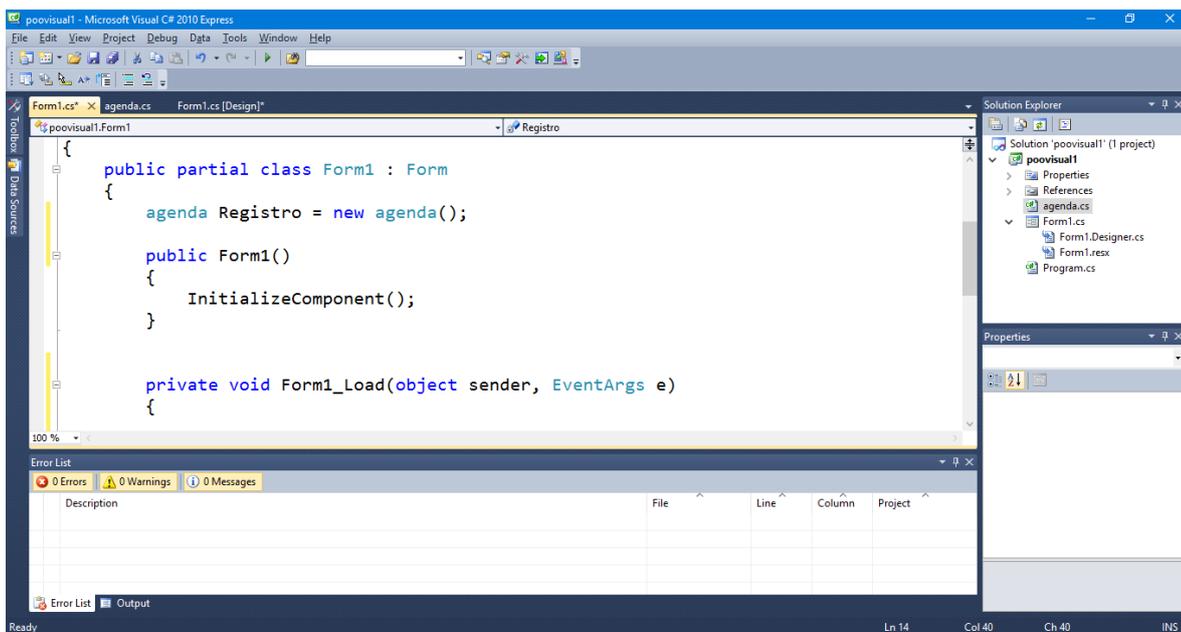
Agora, vamos voltar ao form1 (formulário gráfico). Vamos fazer os objetos acionarem métodos de acordo com algumas ações, que neste caso, serão cliques nos botões.

Como queremos que um objeto seja preenchido para depois ser recuperado durante a execução do programa, precisamos criar uma instância de um objeto do tipo Agenda. Para criarmos a instância que possa ser utilizada em todos os objetos deste formulário, iremos instanciar o objeto de maneira global.

Se clicarmos duas vezes em qualquer lugar do formulário, que não tenha nenhum objeto, apenas o formulário, iremos abrir o método Form\_Load, como na imagem:



Para instanciar um objeto que possa ser utilizado em todo o formulário, iremos fazer isso logo depois da instanciação do Form, como na imagem:

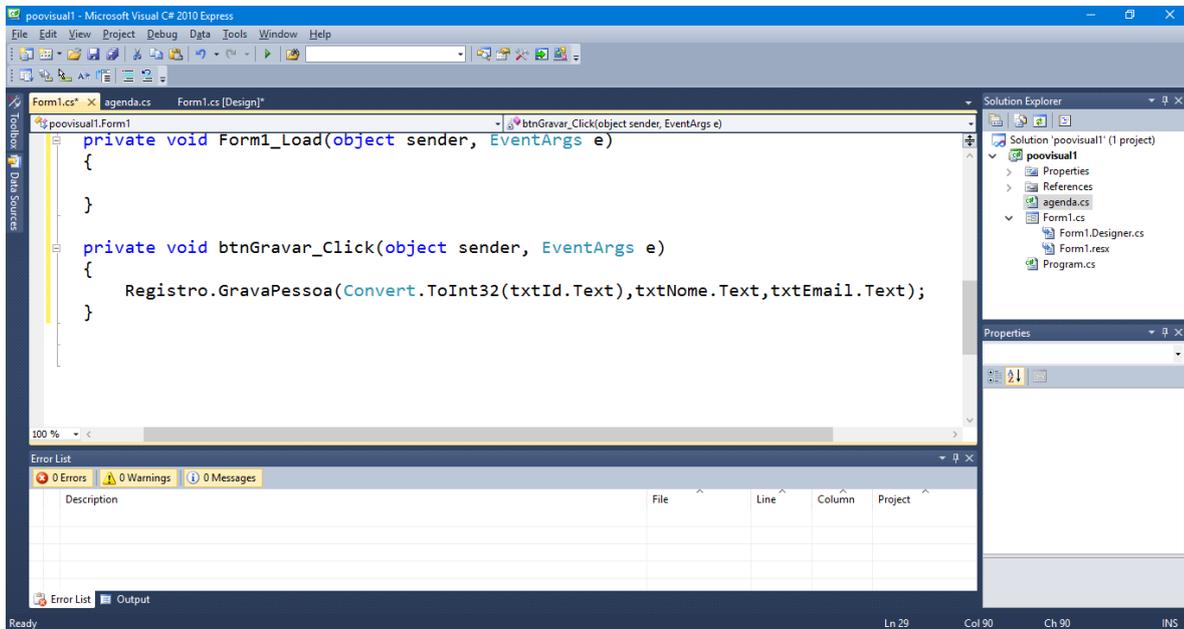


```
agenda Registro = new agenda();
```

Neste caso, criamos um objeto chamado Registro, que é instanciada da classe Agenda (nome da classe que criamos no início).

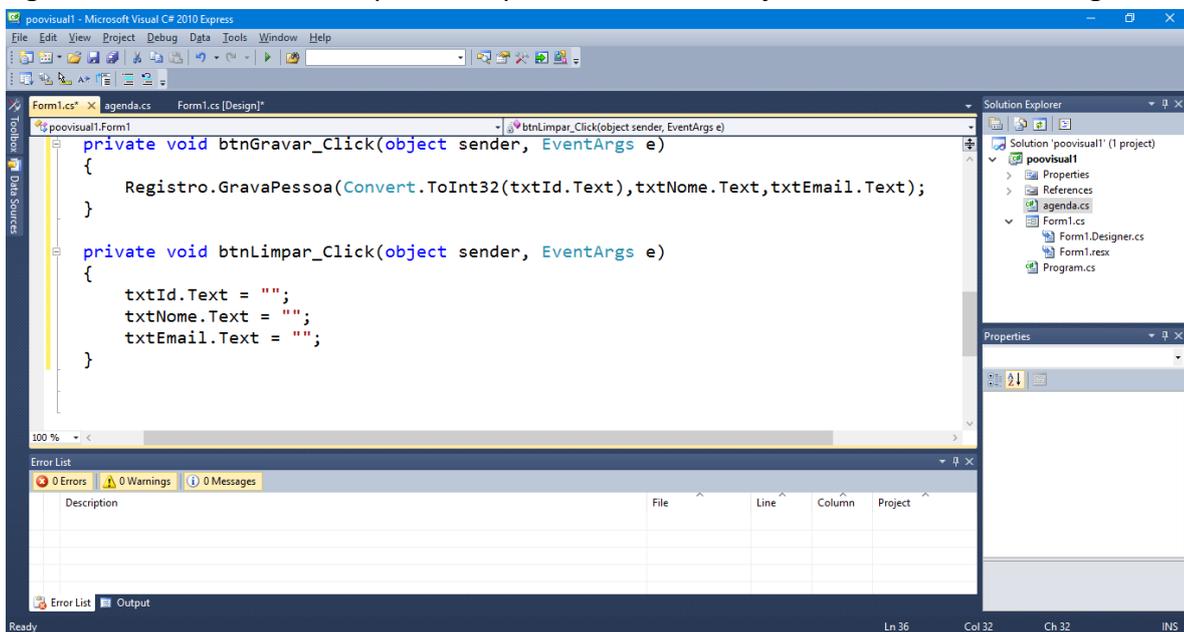
Agora vamos aos Clicks dos botões. Para isso, para criarmos um método Click de um botão de forma fácil, basta dar dois cliques sobre o botão que deseja programar o evento Click. Vamos fazer primeiramente o Gravar Objetos.

Iremos inserir o código:



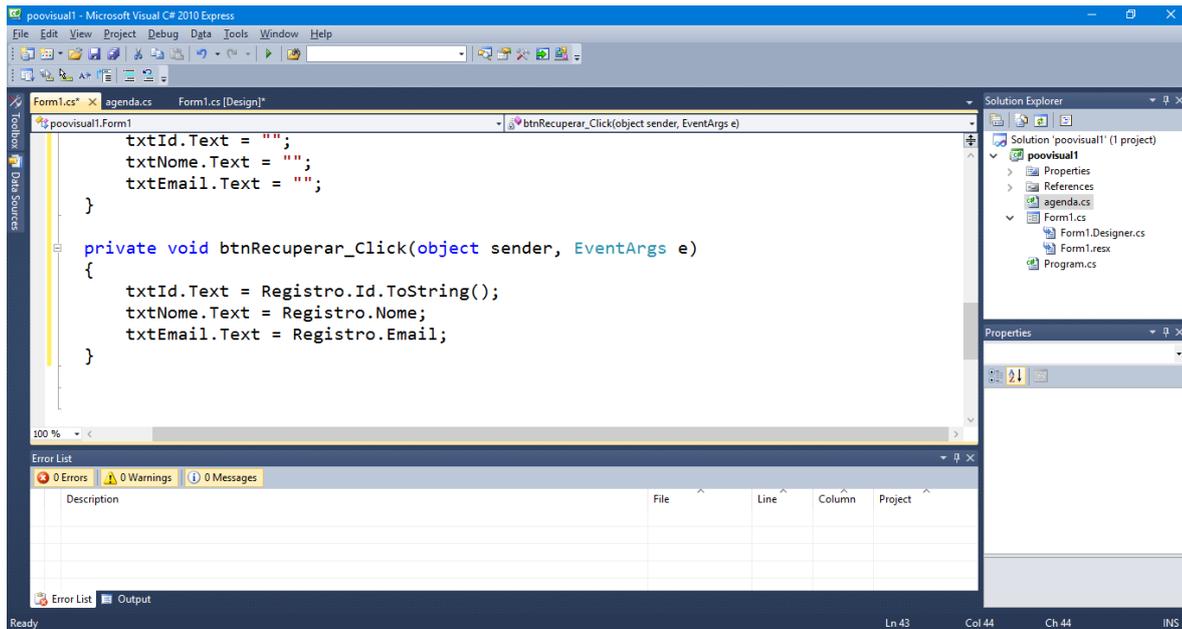
```
private void btnGravar_Click(object sender, EventArgs e)
{
    Registro.GravaPessoa(Convert.ToInt32(txtId.Text),txtNome.Text,txtEmail.Text);
}
```

Agora vamos fazer o botão que irá limpar os dados dos objetos Texts, como na imagem:



```
private void btnLimpar_Click(object sender, EventArgs e)
{
    txtId.Text = "";
    txtNome.Text = "";
    txtEmail.Text = "";
}
```

Por fim, vamos criar o botão que pega os dados de um objeto e coloca nos objetos Texts.



```
private void btnRecuperar_Click(object sender, EventArgs e)  
{  
    txtId.Text = Registro.Id.ToString();  
    txtNome.Text = Registro.Nome;  
    txtEmail.Text = Registro.Email;  
}
```

Pronto. Basta rodar.

Não podemos gravar sem preencher dados. Para isso faça o preenchimento com os dados corretos, pois não utilizamos nenhum tipo de validação.

Exercício:

Crie uma interface chamada Conta que implementa um método Resultado, que recebe como parâmetro dois números do tipo Double, e retorna o resultado de uma operação matemática double da seguinte forma:

```
double resultado(double num1, double num2);
```

Implemente agora quatro classes, que herdam esta interface Conta, uma para cada uma das operações básicas (+, -, / e \*).

Crie um formulário que possui o seguinte formato:

Calculadora

Num1

Num2

+ - \* /

Resultado  
0

São dois textBox para os números 1 e 2 da operação, 4 botões para as operações e 1 label para o resultado, além das label para rotular cada campo.

Ao clicar um dos botões de operação, pegar números digitados nos textBoxes, instanciar o objeto correto para a operação que o botão representa e enviar os parâmetros, recebendo o resultado em lblResultado para exibi-lo.