

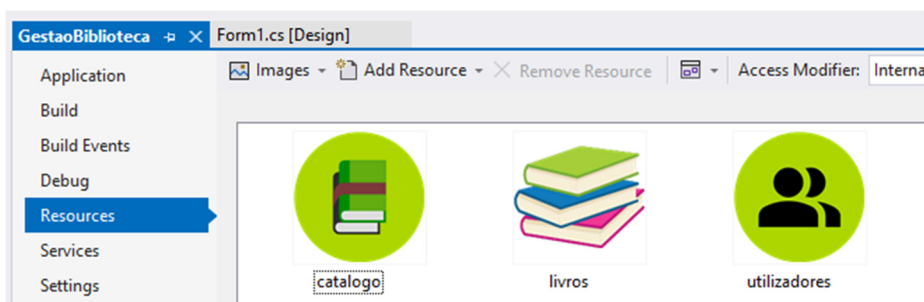
Atividade de Avaliação 2


Criar a primeira classe e primeira form

Abra o projeto **GestaoBiblioteca** criado na atividade anterior.

1. Vamos começar por adicionar as imagens fornecidas pelo professor aos *resources* (recursos) do projeto:

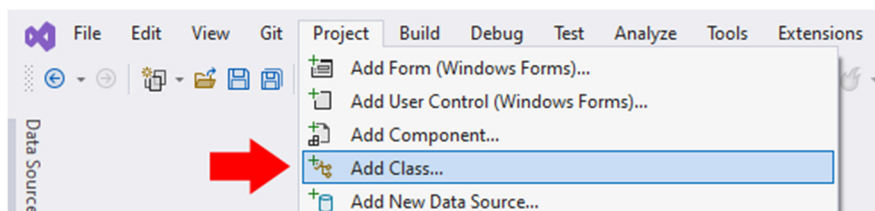
- clique no menu **Project** → **GestaoBiblioteca Properties** / **Projeto** → **Propriedades de GestaoBiblioteca**
- clique em **Resources** / **Recursos**
- clique em **Add Resource** → **Add Existing File...** / **Adicionar Recurso** → **Adicionar Arquivo Existente...**
- adicione os ficheiros



 Faça *commit* com a seguinte mensagem "Adicionar imagens aos recursos do projeto"

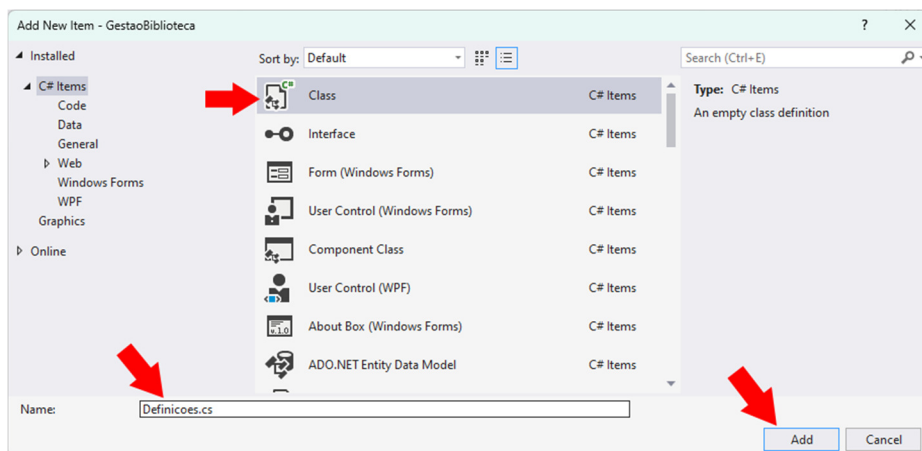
Vamos criar uma classe estática chamada **Definicoes**: esta classe irá ter a responsabilidade de armazenar as definições da aplicação.

2. Para criar a classe **Definicoes**, no menu principal, clique no menu **Project/Projeto**, opção **Add Class...** / **Adicionar classe...**:



Na janela **Add New Item/Adicionar Novo Item**, especifique o nome **Definicoes.cs**:

Atividade de Avaliação 2



Vamos transformar a classe **Definicoes** numa classe estática. Para tal, basta acrescentar a palavra **static** à definição da classe:

A cinzento encontra-se o código já existente: não modifique esse código

```
namespace GestaoBiblioteca
{
    internal static class Definicoes
    {
    }
}
```

Adicione uma constante chamada **NomeAplicacao**. Esta constante armazena o nome da aplicação e permite que a qualquer momento se altere o nome da mesma, sendo depois a alteração refletida na interface (**a cinzento encontra o código já existente que não deve alterar**):

```
internal static class Definicoes
{
    public const string NomeAplicacao = "Bibliotex";
}
```

Neste caso foi definido o nome **Bibliotex**. Utilize este nome ou escolha outro ao seu gosto.

Faça **commit** com a seguinte mensagem "Adicionar classe Definicoes"

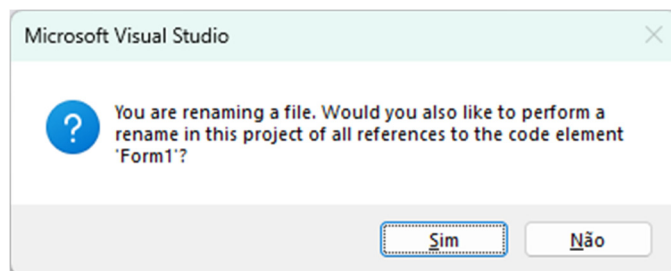
Vamos agora mudar o nome da **Form1** para **FormPrincipal**:

3. Comece por fechar a janela **Form1** (se estiver aberta) e mude o nome para **FormPrincipal** clicando com o botão direito em **Form1.cs** e clicando na opção **Rename/Renomear**:



Surge uma janela de confirmação, perguntando se deseja mudar todas as referências no projeto. Responda **Yes/Sim**:

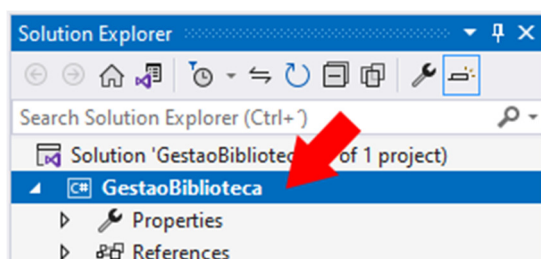
Atividade de Avaliação 2



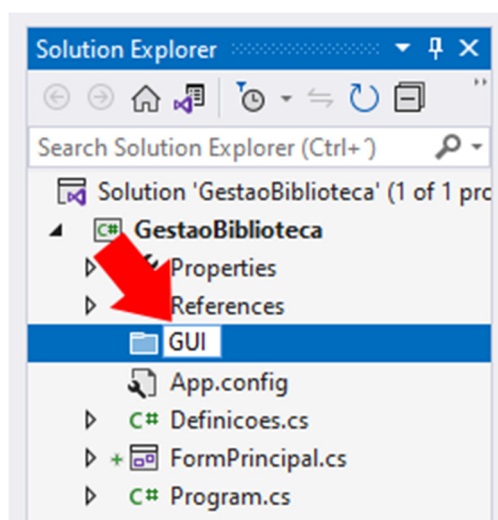
Faça *commit* com a seguinte mensagem "Mudar o nome da Form1 para FormPrincipal"

Vamos agora criar uma pasta onde irão ser colocadas todas as *forms* deste projeto.

4. No **Solution Explorer**/Gerenciador de Soluções clique com o botão direito do rato no nome do projeto e clique na opção **Add → New Folder/Adicionar → Nova Pasta**:

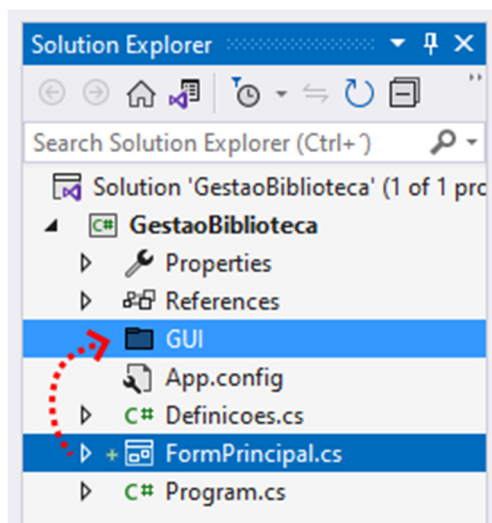


Atribua à nova pasta o nome **GUI**:

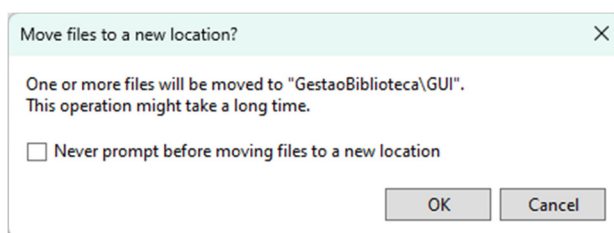


5. Mova a **FormPrincipal** para dentro da pasta **GUI**, arrastando-a com o botão esquerdo do rato:

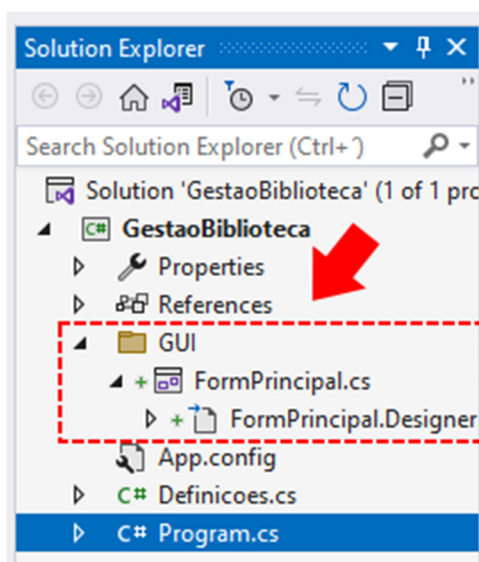
Atividade de Avaliação 2




Irá surgir uma janela de confirmação: clique em OK



Resultado final:



 Faça *commit* com a seguinte mensagem "Mover a FormPrincipal para pasta GUI"

Atividade de Avaliação 2

6. Crie a *form* de autenticação `FormAutenticacao`: esta *form* é utilizada como ponto de entrada na aplicação, devendo o utilizador introduzir o seu nome de utilizador e password.

Para adicionar a `FormAutenticacao` ao projeto, clique com o botão direito do rato em cima da pasta `GUI` e clique na opção `Add → Form (Windows Forms)... / Adicionar → Formulário (Windows Forms)...`

Após criar a *form* e antes de adicionar os controlos faça *commit* com a seguinte mensagem "Adicionar `FormAutenticacao`"

7. Vamos agora fazer com que a *forma* de autenticação seja chamada quando o programa entra em execução. Para tal modifique o ficheiro `Program.cs`:

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);

    // Efetuar autenticação
    Application.Run(new GUI.FormAutenticacao());
}
```

Faça *commit* com a seguinte mensagem "Executar autenticação"

8. Voltando à `FormAutenticacao`, modifique as seguintes propriedades da *form*:

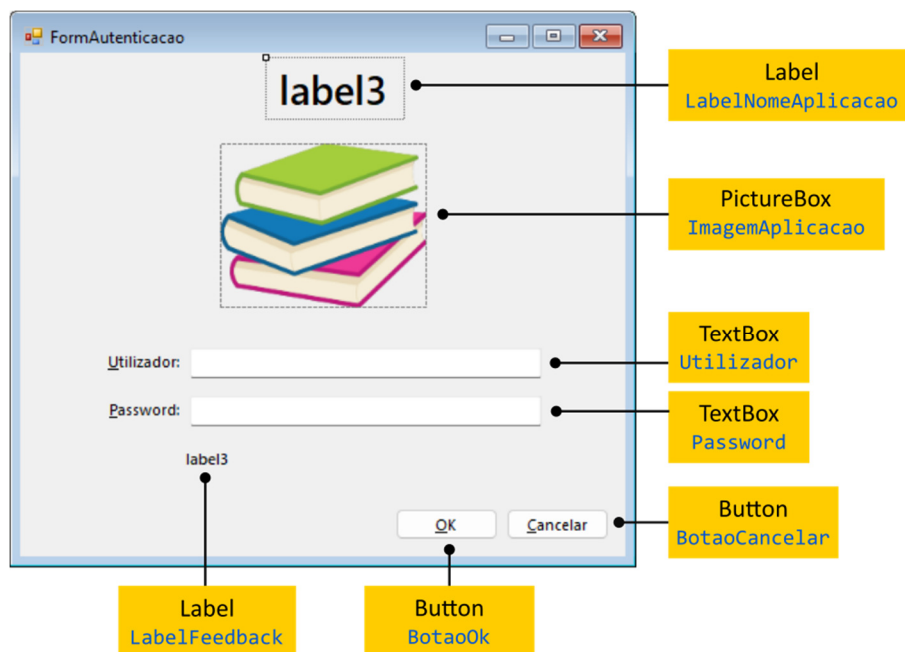
Tarefa	Propriedade
Tipo de letra (de toda a <i>form</i>)	Font: Segoe UI; 8,25pt

Atividade de Avaliação 2

Dimensões da janela (width; height)

Size: 458; 430

9. Adicione os controlos necessários à *form*, não esquecendo de especificar a *access key*:



Faça *commit* com a seguinte mensagem "Adicionar controlos à FormAutenticacao"

10. Defina a *tab order* (ordem de tabulação) dos controlos da *form* de acordo com a seguinte imagem:



Faça *commit* com a seguinte mensagem "Definir tab order da FormAutenticacao"

Atividade de Avaliação 2

11. Abra o código da *form* e acrescente o seguinte código logo no início da definição da classe *FormAutenticacao*:

```
public partial class FormAutenticacao : Form
{
    /// <summary>
    /// Indica se o utilizador está autenticado ou não.
    /// </summary>
    public bool Autenticado { get; private set; } = false;

    public FormAutenticacao()
    {
```

12. Adicione o seguinte código ao construtor:

```
public FormAutenticacao()
{
    InitializeComponent();

    // A janela deverá surgir no centro do ecrã
    this.StartPosition = FormStartPosition.CenterScreen;
}
```

13. Adicione o evento *Load()* com o seguinte código para inicialização da *form*:

```
private void FormAutenticacao_Load(object sender, EventArgs e)
{
    this.SuspendLayout();

    this.AcceptButton = BotaoOk;
    this.CancelButton = BotaoCancelar;
    this.Text = "Autenticação";
    this.FormBorderStyle = FormBorderStyle.FixedDialog;
    this.MinimizeBox = false;
    this.MaximizeBox = false;

    LabelNomeAplicacao.Text = Definicoes.NomeAplicacao;

    // Posicionar controlos
    ImagemAplicacao.Left = (this.ClientSize.Width / 2) - (ImagemAplicacao.Width / 2);
    LabelNomeAplicacao.Left = (this.ClientSize.Width / 2) - (LabelNomeAplicacao.Width / 2);

    LabelFeedback.Visible = false;
    Password.UseSystemPasswordChar = true;

    this.ResumeLayout(false);
}
```




O código acima efetua algumas operações de inicialização sem que seja necessário recorrer ao componente Form Designer. Pode também ser facilmente copiado e colado noutras *forms*, acelerando o tempo de desenvolvimento da aplicação.

Analise o código de inicialização e tente perceber todas as definições que foram alteradas.




Faça *commit* com a seguinte mensagem "Inicializar FormAutenticacao"

14. Insira código no evento `click` do botão `BotaoCancelar` para terminar a aplicação.

 Faça *commit* com a seguinte mensagem "Cancelar FormAutenticacao"

15. Insira código no evento `click` do botão `BotaoOk` para fechar a janela (este código é temporário até que seja efetivamente efetuada a autenticação):

```
/// <summary>
/// Efetuar a autenticação.
/// </summary>
private void BotaoOk_Click(object sender, EventArgs e)
{
    Autenticado = true;
    this.Close();
}
```

 Faça *commit* com a seguinte mensagem "Implementar autenticação rudimentar"

COTAÇÕES

Questão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pontos	15	18	10	5	10	10	10	10	55	13	5	5	13	10	11

Total: 200 pontos