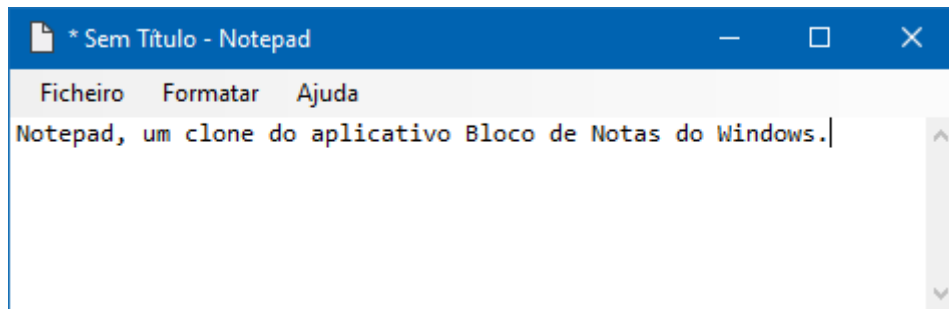


Ficha de Trabalho

Desenvolvimento de um editor de texto

O objetivo desta ficha de trabalho é criar um pequeno aplicativo utilizando a linguagem C# e a tecnologia Windows Forms. O aplicativo é um editor de texto simples, à semelhança do Bloco de Notas do sistema operativo Windows.

Este documento contém as instruções que deverá seguir para atingir o resultado final.



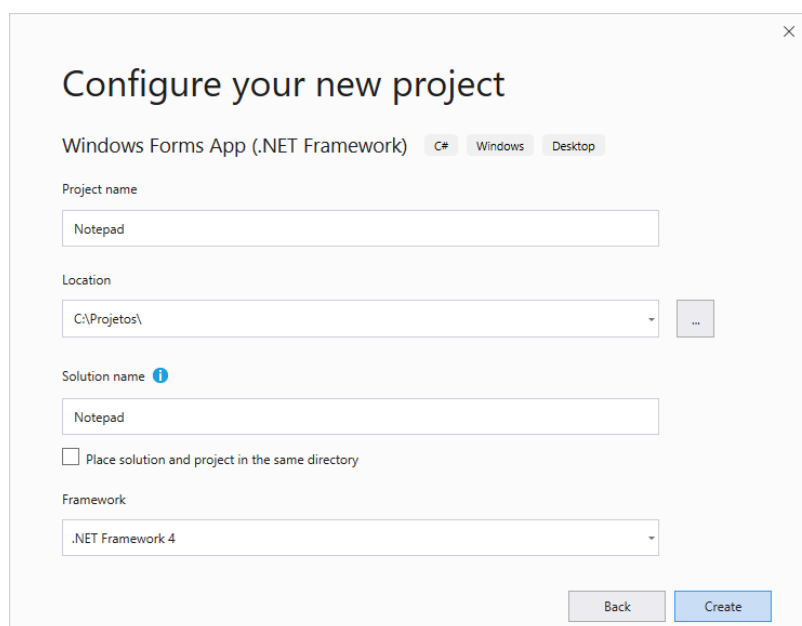
Nota

As instruções e as imagens contidas neste documento dizem respeito ao editor Visual Studio Community Edition 2019. No entanto este projeto pode ser facilmente desenvolvido utilizando versões mais antigas do editor Visual Studio.





GRUPO I

Funcionalidades Básicas

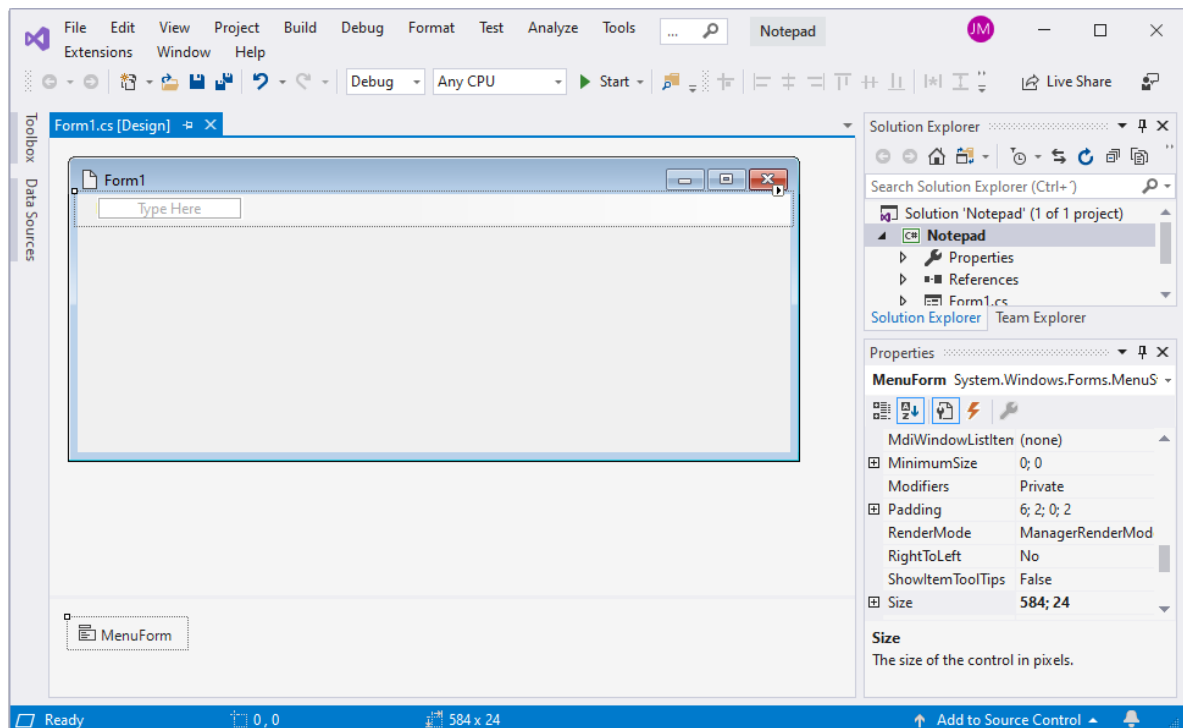
1. Crie um novo projeto Windows Forms intitulado **Notepad**:



2. Vamos começar por modificar algumas propriedades da janela principal da aplicação:

Tarefa	Propriedade	
Atribuir um ícone à aplicação	Icon: Notepad.ico	
Definir um tipo de letra para toda a janela	Font: SegoeUI	
Posicionamento inicial da janela no centro do ecrã	StartPosition: CenterScreen	
Dimensões da janela (width; height)	Size: 600; 250	

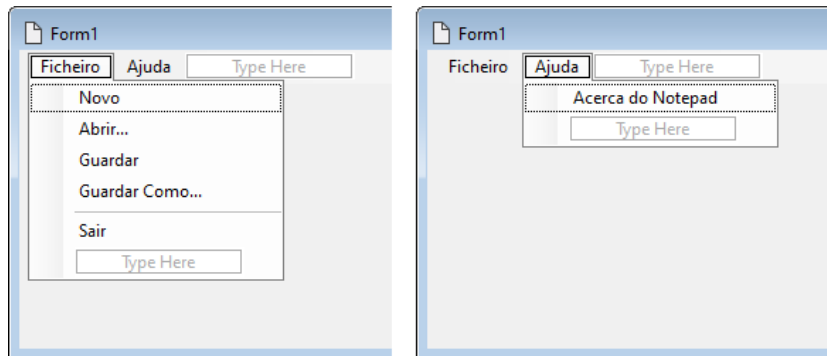
3. Insira um novo controlo **MenuStrip** e mude o nome do mesmo para **MenuForm**:



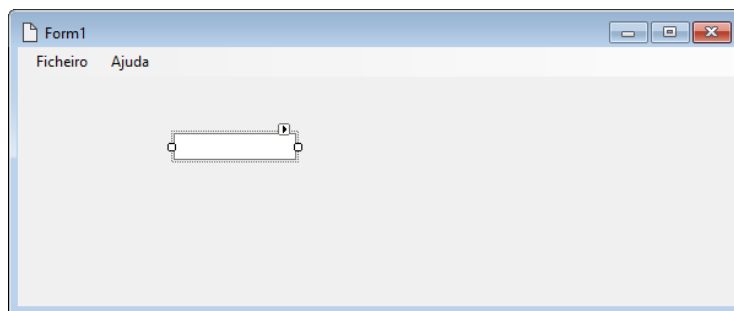
4. Insira os seguintes itens no menu:

Menu	Itens do menu	Atalho (<i>shortcut</i>)	Nome
<u>E</u> icheiro		F	
	<u>N</u> ovo	N	MenuFormNovo
	<u>A</u> brir...	A	MenuFormAbrir
	<u>G</u> uardar	G	MenuFormGuardar
	Guardar <u>C</u> omo...	C	MenuFormGuardarComo
	-		
	<u>S</u> air	S	MenuFormSair
<u>A</u> jud		A	
	<u>A</u> cerca do Notepad	A	MenuFormAcerca

Após a inserção dos menus, a janela **Form1** deverá ter um aspeto semelhante às seguintes imagens:



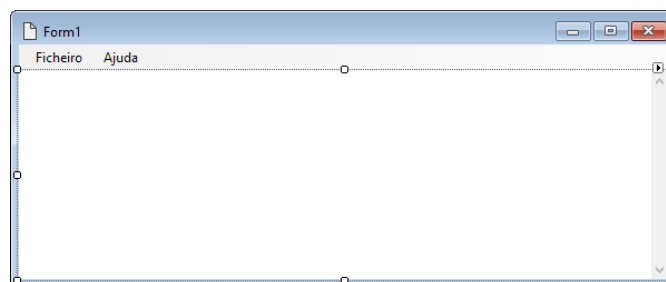
5. Insira um novo controlo **TextBox** (caixa de texto) na janela:



6. Modifique as seguintes propriedades da caixa de texto:

Tarefa	Propriedade
Nome do controlo	Name: Texto
Permitir que o utilizador possa pressionar a tecla TAB dentro da caixa de texto	AcceptsTab: True
Retirar a border à caixa de texto	BorderStyle: None
Permitir que o controlo preencha todo o espaço possível da janela	Dock: Fill
Definir o tipo de letra para a caixa de texto	Font: Consolas, 10pt
Aumentar o número máximo de caracteres que a caixa de texto aceita	MaxLength: 65535
A caixa de texto contém várias linhas de texto	Multiline: True
Adicionar scrollbars (barras de deslocamento) à caixa de texto	ScrollBars: Both

Se as propriedades foram corretamente modificadas, a caixa de texto deverá preencher toda a janela principal (com exceção da área destinada ao menu):



7. Efetue duplo clique no barra de título da janela para aceder ao código da janela. Na vista de código (do ficheiro `Form1.cs`), insira o seguinte código para importar o namespace `System.IO`:

```
7 using System.Text;
8 using System.Windows.Forms;
9 using System.IO;
10
```

8. No evento `Form1_Load()` insira o seguinte código para atribuir um título inicial à janela da aplicação:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "Sem Título - Notepad";
}
```

9. Vamos agora inserir código para terminar a aplicação, quando o utilizador clica no item `Sair` do menu `Ficheiro`. Para tal insira o seguinte código no evento `Click()` do item `MenuFormSair`:

```
/// <summary>
/// Terminar a aplicação.
/// </summary>
private void MenuFormSair_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Evento `Click()`

Para criar o evento `Click()` para um item de um menu:

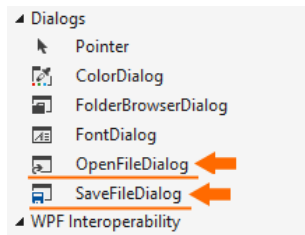
1. Aceda à vista de Design
2. Efetue duplo-clique sobre o item pretendido
3. A estrutura básica do evento é criada automaticamente

10. De seguida vamos inserir código para poder criar um novo ficheiro, quando o utilizador clica no item `Novo` do menu `Ficheiro`:

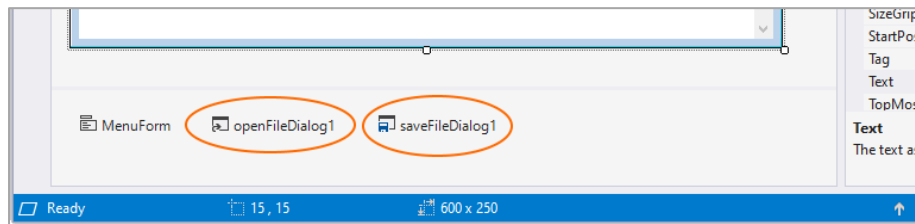
```
/// <summary>
/// Criar um novo ficheiro.
/// </summary>
private void MenuFormNovo_Click(object sender, EventArgs e)
{
    // Eliminar o conteúdo da caixa de texto
    Texto.Text = string.Empty;

    // Inicialmente o novo ficheiro não tem nome
    this.Text = "Sem título - Notepad";
}
```

11. A aplicação deverá poder abrir e gravar ficheiros. Para tal é necessário fornecer ao utilizador um mecanismo para navegar no sistema de ficheiros do computador, gravar ficheiros e abrir ficheiros. Para executar estas operações são utilizadas janelas de diálogo (*dialog windows*). Insira os seguintes controlos do tipo `Dialog`: `OpenFileDialog` e `SaveFileDialog` (categoria `Dialog` na `Toolbox`).



Após a inserção dos controlos, estes deverão aparecer automaticamente na zona inferior da janela, junto ao controlo `MenuForm`:



12. Para gerir o ficheiro que está em edição num dado momento, será necessário adicionar as propriedades privadas `nomeFicheiro` e `localFicheiro`, as quais irão armazenar o nome do ficheiro em edição e o respetivo local onde está armazenado. Adicione o seguinte código, no início da definição da classe `Form1`:

Atenção

O texto a cinzento corresponde a instruções que já se encontram no código. Não as modifique: insira apenas as novas instruções.

```
public partial class Form1 : Form
{
    /// <summary>
    /// O nome do ficheiro em edição.
    /// </summary>
    private string nomeFicheiro = string.Empty;

    /// <summary>
    /// A localização do ficheiro em edição.
    /// </summary>
    private string localFicheiro = string.Empty;

    public Form1()
    {
        InitializeComponent();
    }
}
```

13. Quando o utilizador clica na opção `Novo` é necessário limpar o valor das propriedades `nomeFicheiro` e `localFicheiro`. Acrescente o seguinte código a meio do evento `MenuFormNovo_Click()`:

```

/// <summary>
/// Criar um novo ficheiro.
/// </summary>
private void MenuFormNovo_Click(object sender, EventArgs e)
{
    // Eliminar o conteúdo da caixa de texto
    Texto.Text = string.Empty;

    // Limpar as informações do ficheiro
    nomeFicheiro = string.Empty;
    localFicheiro = string.Empty;

    // Inicialmente o novo ficheiro não tem nome
    this.Text = "Sem título - Notepad";
}

```

14. Vamos agora permitir que o utilizador abra um ficheiro de texto. Para tal vamos inserir o seguinte código quando o utilizador clica na opção **Abrir** do menu **Ficheiro**:

```

/// <summary>
/// Abrir um novo ficheiro.
/// </summary>
private void MenuFormAbrir_Click(object sender, EventArgs e)
{
    // Título
    openFileDialog1.Title = "Abrir";
    // Permitir selecionar apenas um ficheiro
    openFileDialog1.Multiselect = false;
    // Tipos de ficheiros que podem ser abertos
    openFileDialog1.Filter = "Ficheiros de texto (*.txt)|*.txt|Todos os ficheiros (*.*)|*.*";
    openFileDialog1.FilterIndex = 0;

    // Verificar que o nome e a localização especificadas pelo utilizador existem
    openFileDialog1.CheckFileExists = true;
    openFileDialog1.CheckPathExists = true;

    openFileDialog1.FileName = string.Empty;

    // Mostrar o OpenFileDialog e verificar o resultado da operação
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Armazenar o nome e a localização do novo ficheiro
        nomeFicheiro = Path.GetFileName(openFileDialog1.FileName);
        localFicheiro = Path.GetDirectoryName(openFileDialog1.FileName);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";

        // Abrir o ficheiro e colocar o conteúdo na caixa de texto
        Texto.Text = File.ReadAllText(openFileDialog1.FileName, Encoding.UTF8);
    }
}

```

15. É necessário criar os seguintes métodos:

- **Guardar()**: efetua a gravação de um ficheiro
- **GuardarComo()**: efetua a gravação de um ficheiro num determinado local e com um determinado nome, escolhidos pelo utilizador

Estas operações são implementadas através de dois métodos e não diretamente nos eventos `Click()` das opções `Guardar` e `Guardar Como` porque serão chamados a partir de diferentes pontos do programa. Por exemplo, se um ficheiro nunca foi gravado, é executado o método `GuardarComo()`, caso contrário é executado o método `Guardar()`.

Copie e cole o seguinte código:

```

/// <summary>
/// Guardar o ficheiro atual.
/// </summary>
private void Guardar()
{
    // Se o ficheiro ainda não foi guardado, executar o método GuardarComo()
    if ((nomeFicheiro == string.Empty) || (localFicheiro == string.Empty))
    {
        GuardarComo();
    }
    else
    {
        // Guardar o ficheiro
        File.WriteAllText(Path.Combine(localFicheiro, nomeFicheiro), Texto.Text,
            Encoding.UTF8);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";
    }
}

/// <summary>
/// Guardar o ficheiro atual especificando um nome e localização.
/// </summary>
private void GuardarComo()
{
    saveFileDialog1.Title = "Guardar como";
    saveFileDialog1.Filter = "Ficheiros de texto (*.txt)|*.txt|Todos os ficheiros (*.*)|*.*";
    saveFileDialog1.FilterIndex = 0;
    saveFileDialog1.DefaultExt = ".txt";

    // Verificar que a localização especificada pelo utilizador existe
    saveFileDialog1.CheckPathExists = true;

    // Ir para a diretoria do ficheiro (se existir uma localização de ficheiro)
    if (localFicheiro != string.Empty)
    {
        saveFileDialog1.InitialDirectory = localFicheiro;
    }

    // Especificar o nome do ficheiro
    saveFileDialog1.FileName = nomeFicheiro;

    // Mostrar o SaveFileDialog e verificar o resultado da operação
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Armazenar o nome e a localização na qual o ficheiro foi gravado
        nomeFicheiro = Path.GetFileName(saveFileDialog1.FileName);
        localFicheiro = Path.GetDirectoryName(saveFileDialog1.FileName);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";

        // Gravar o conteúdo do ficheiro
        File.WriteAllText(saveFileDialog1.FileName, Texto.Text, Encoding.UTF8);
    }
}

```

```
}
}
```

16. Insira o seguinte código no evento `Click()` dos itens `MenuFormGuardar` e `MenuFormGuardarComo`:

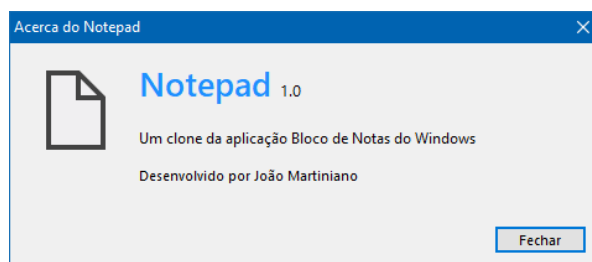
```
/// <summary>
/// Guardar o ficheiro atual.
/// </summary>
private void MenuFormGuardar_Click(object sender, EventArgs e)
{
    Guardar();
}

/// <summary>
/// Guardar o ficheiro atual, especificando um nome e localização.
/// </summary>
private void MenuFormGuardarComo_Click(object sender, EventArgs e)
{
    GuardarComo();
}
```

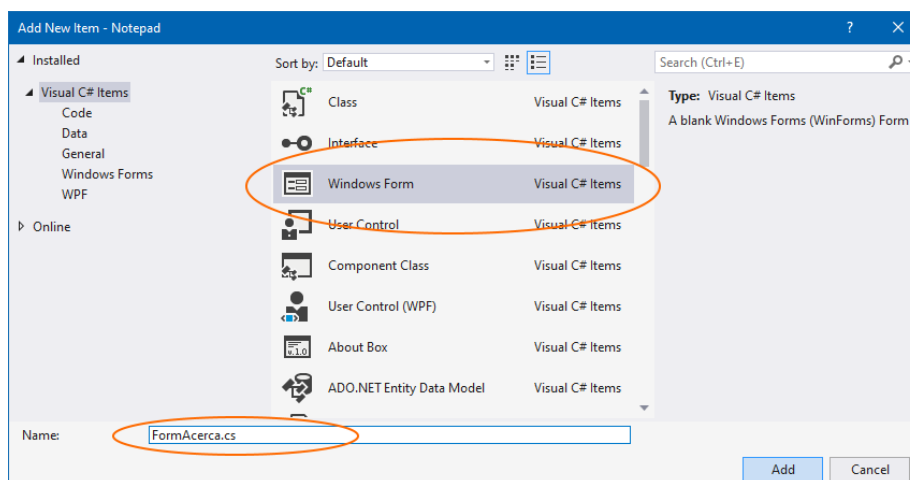
GRUPO II

About Form

17. Qualquer aplicação necessita de uma *about form*, que mostra informações acerca da mesma. Nos pontos seguintes será criada uma janela deste tipo, igual à seguinte imagem:



Para tal crie uma nova janela, com o nome de ficheiro `FormAcerca.cs` (clique em `Project` → `Add Windows Form...`):

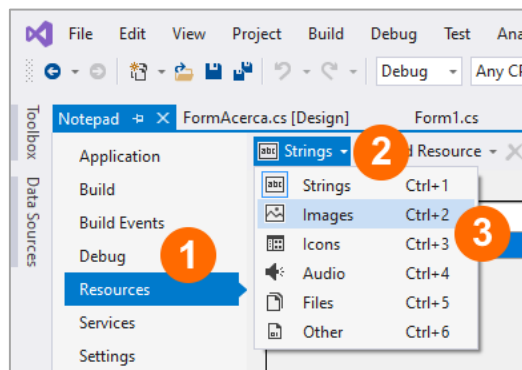



18. Modifique algumas propriedades da janela:

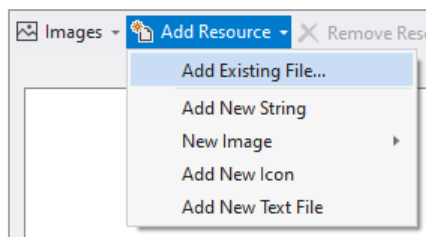
Tarefa	Propriedade
Definir um tipo de letra para toda a janela	Font: SegoeUI
Definição da border	FormBorderStyle: FixedDialog
Retirar o ícone para maximizar a janela	MaximizeBox: False
Retirar o ícone para minimizar a janela	MinimizeBox: False
Não mostrar o ícone na barra de tarefas do Windows (<i>taskbar</i>)	ShowInTaskbar: False
Dimensões da janela (width; height)	Size: 494; 221
Posicionamento inicial da janela no centro da janela da aplicação	StartPosition: CenterParent
Título da janela	Text: "Acerca do Notepad"


19. De modo a poder mostrar uma imagem na janela (o ícone da aplicação), vamos criar um *resource* e importar a imagem. Para tal siga as seguintes instruções:

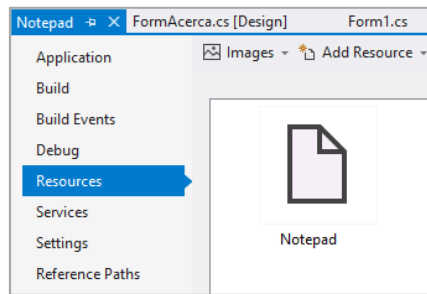
- Clique no menu [Project](#), opção [Notepad Properties...](#)
- Clique no separador [Resources](#) (1), depois clique na lista do item [Strings](#) (2) e seleccione a opção [Images](#) (3)



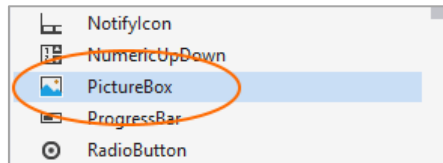
- Clique no ícone  posicionado no lado direito da opção [Add Resource](#) e clique depois em [Add Existing File...](#):



- Selecione o ficheiro [Notepad.png](#) (fornecido juntamente com esta ficha) e clique em [Abrir](#)
- Se a operação tiver sido executada corretamente a imagem fará agora parte dos recursos do projeto (não se esqueça de gravar, clicando no ícone )



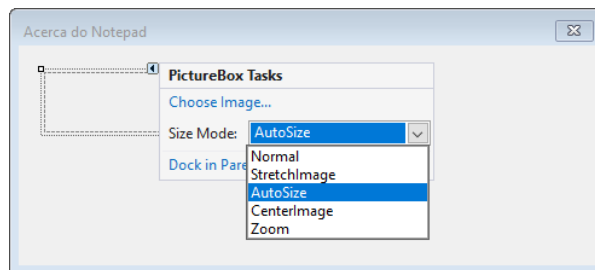
20. Insira um controlo **PictureBox** (caixa de imagem):



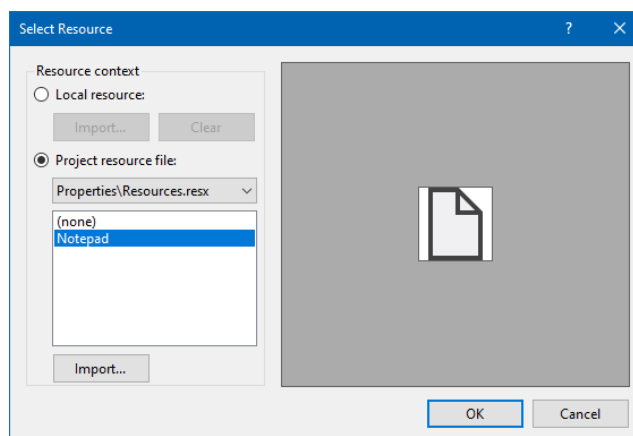
21. Posicione o controlo **PictureBox** nas coordenadas ($X = 21$, $Y = 22$). Para tal modifique a propriedade **Location**.

22. Clique no ícone  do controlo **PictureBox** e efetue as seguintes operações:

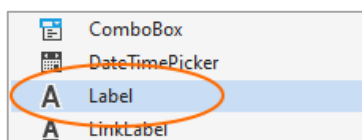
- a. Na opção **SizeMode** selecione o valor **AutoSize**:



- b. Clique na opção **Choose Image...** e selecione a imagem **Notepad** a partir dos recursos do projeto:

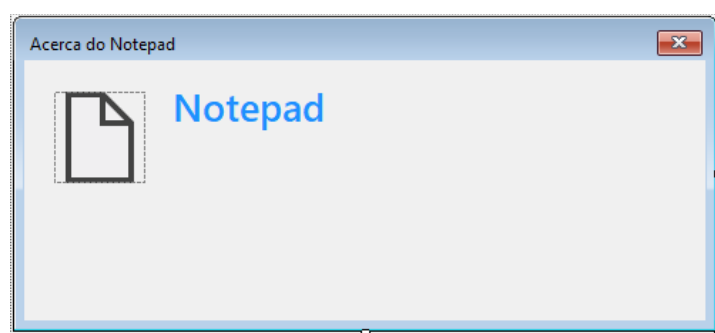


23. Insira um controlo **Label** e mude as seguintes propriedades do mesmo:

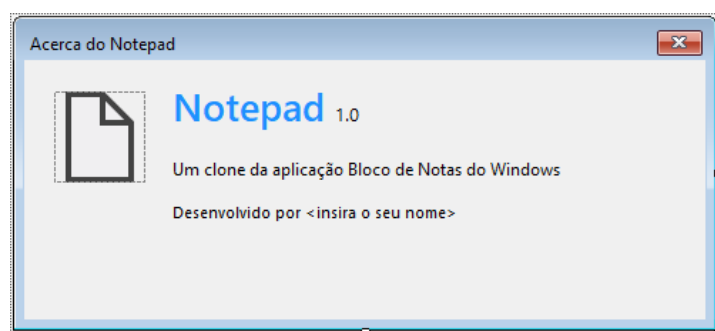


Tarefa	Propriedade
Definir o tipo de letra e o tamanho	Fonte: SegoeUI Estilo: Semi Negrito Tamanho: 20
Cor de carater	ForeColor: DodgerBlue
Texto	Text: Notepad

24. Tente posicionar o controlo **Label** de acordo com a seguinte imagem:

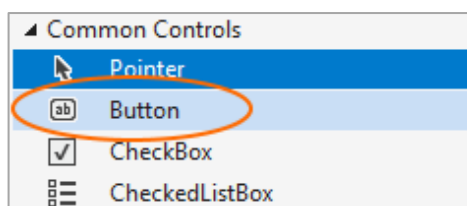


25. Insira três controlos **Label**, posicione-os e insira o respetivo texto, conforme a imagem seguinte:

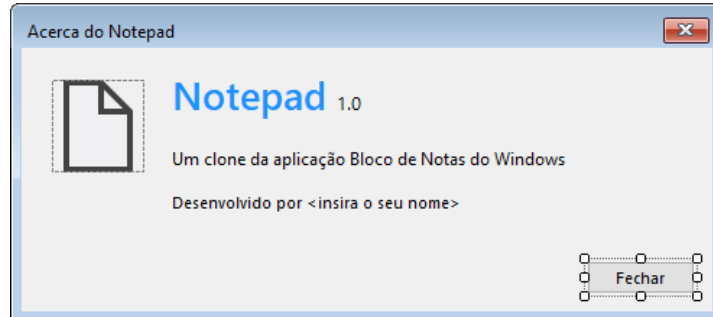


Conteúdo do label	Descrição	Deverá fazer...
"1.0"	Versão da aplicação	Formatar o tamanho da letra: 10
"Um clone da aplicação..."	Descrição da aplicação	Formatar o tamanho da letra: 9
"Desenvolvido por..."	Autor da aplicação	Insira o seu nome

26. Insira um controlo **Button** (botão), com as seguintes definições:



Tarefa	Propriedade
Nome do controlo	Name: <code>BtFechar</code>
Posicionamento na janela	Location: <code>391; 137</code>
Texto	Text: <code>Fechar</code>



27. Faça duplo clique sobre o botão de modo a criar o evento `Click()` e insira o seguinte código para fechar a janela `Acerca` e voltar à janela principal da aplicação:

```
/// <summary>
/// Fechar esta janela.
/// </summary>
private void BtFechar_Click(object sender, EventArgs e)
{
    this.Close();
}
```

28. Falta apenas uma coisa para poder mostrar a janela `Acerca`. Vamos inserir o seguinte código no evento `Click()` do item `Acerca do Notepad` (menu `Ajuda`):

```
/// <summary>
/// Mostrar informações acerca desta aplicação.
/// </summary>
private void MenuFormAcerca_Click(object sender, EventArgs e)
{
    FormAcerca frmAcercaNotepad = new FormAcerca();

    frmAcercaNotepad.ShowDialog();
    frmAcercaNotepad.Dispose();
}
```

Basicamente o que este código faz é:

- Criar uma nova instância da form `FormAcerca`: o nome da instância é `frmAcercaNotepad`
- Mostra a janela `Acerca` em modo *dialog*: ou seja, transfere o controlo da execução do programa para essa janela; a janela principal apenas volta a assumir o controlo quando é fechada a janela `Acerca`
- Após ser fechada a janela `Acerca`, é descartada a instância e todos os recursos que esta utilizou

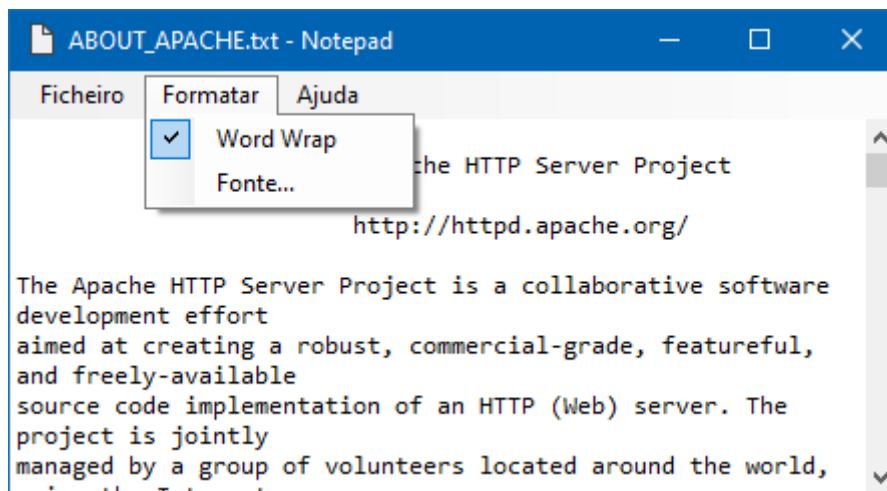
GRUPO III

Word Wrap e Tipo de Letra

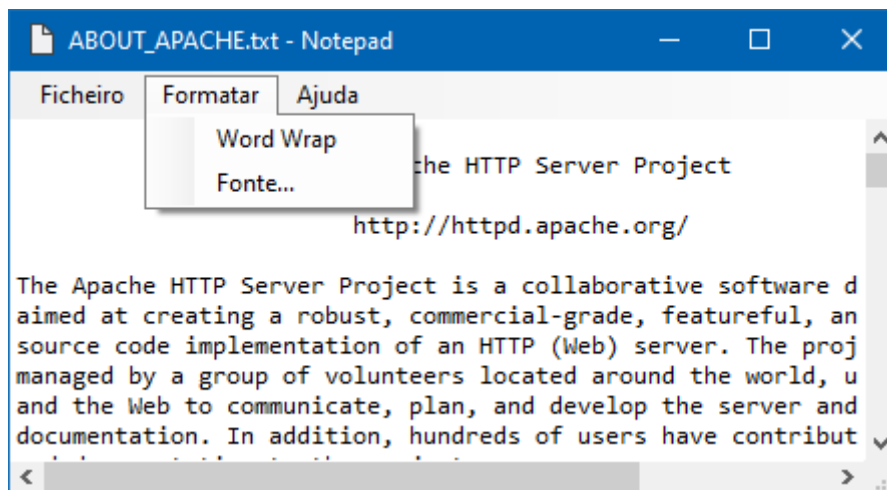
Neste grupo vamos implementar uma funcionalidade que permite mostrar o texto com ou sem *word wrap*:

- Quando a funcionalidade *word wrap* está ativa, as linhas de texto que ultrapassam a largura da janela, continuam na linha seguinte
- Quando a funcionalidade *word wrap* está inativa, o utilizador necessita de efetuar *scroll* para a direita (utilizando a barra de deslocamento horizontal) para visualizar as linhas de texto

Com *word wrap*



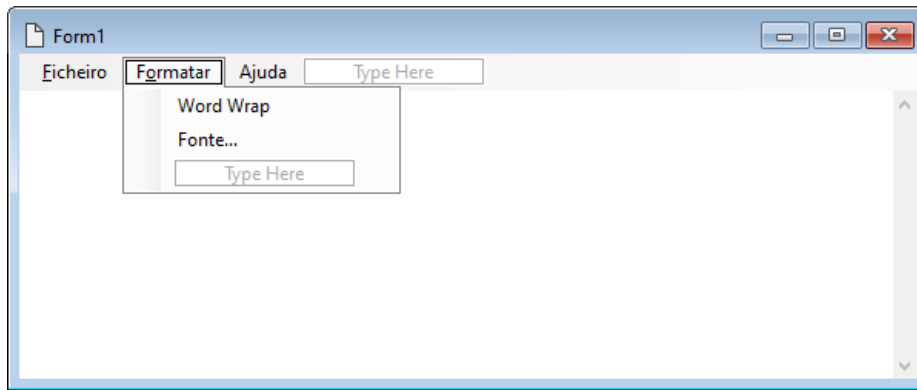
Sem *word wrap*



A funcionalidade *word wrap* irá ficar acessível a partir de um novo menu, intitulado **Formatar**. Este menu irá também conter o item **Fonte**, o qual irá permitir mudar o tipo de letra do texto.

29. Crie o menu **Formatar** e os seguintes itens:

Menu	Itens do menu	Atalho (<i>shortcut</i>)	Nome
Formatar		O	
	Word Wrap	W	MenuFormWordWrap
	Fonte	F	MenuFormFonte



30. Acrescente no evento `Load()` o seguinte código para inicializar a funcionalidade *word wrap*:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "Sem Título - Notepad";

    // Inicializar a opção Word Wrap
    Texto.WordWrap = true;
    // Assinalar na opção do menu Formatar, inserindo um checkmark, que a opção
    Word Wrap está ativa
    MenuFormWordWrap.Checked = true;
}
```

31. Insira o seguinte código no evento `Click()` do item *Word Wrap*:

```
/// <summary>
/// Ligar/desligar word wrap na caixa de texto.
/// </summary>
private void MenuFormWordWrap_Click(object sender, EventArgs e)
{
    // Atribuir à propriedade WordWrap o valor contrário ao que tem neste momento
    Texto.WordWrap = !Texto.WordWrap;

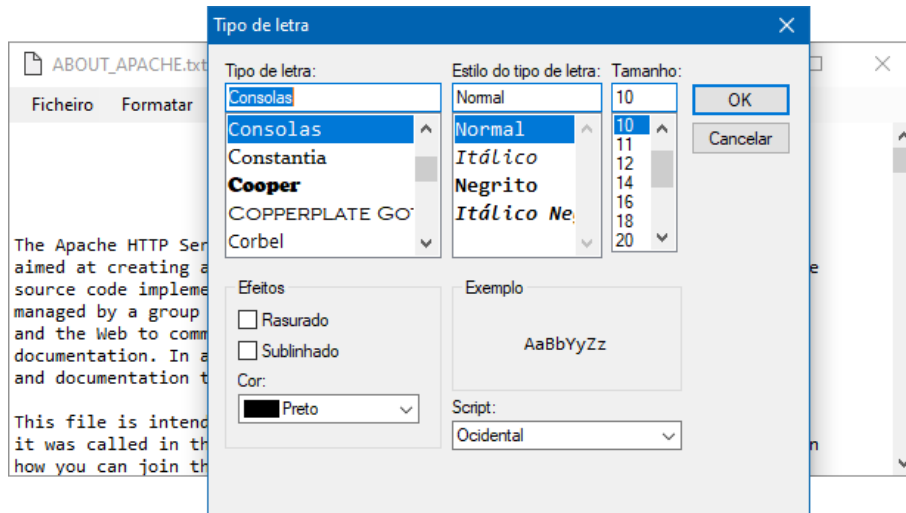
    // Mostrar/esconder uma checkmark consoante o valor da propriedade WordWrap
    MenuFormWordWrap.Checked = Texto.WordWrap;
}
```

Se as indicações anteriores tiverem sido implementadas corretamente, a funcionalidade *word wrap* deverá executar corretamente.

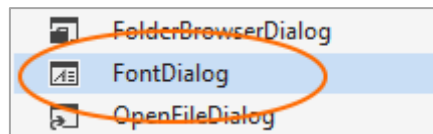
Vamos agora permitir que o utilizador mude alguma formatação do texto, nomeadamente:

- o tipo de letra
- o estilo do tipo de letra (normal, itálico, negrito, etc.)
- o tamanho
- efeitos
- a cor da letra

Esta funcionalidade é muito simples de implementar e é conseguida utilizando um controlo `FontDialog`:



32. Comece por adicionar o controlo `FontDialog`:



33. Insira o seguinte código no evento `Click()` do item `Word Wrap`:

```
/// <summary>
// Mostrar ao utilizador uma caixa de diálogo para escolher o tipo de letra.
/// </summary>
private void MenuFormFonte_Click(object sender, EventArgs e)
{
    // Inicializar a fonte e a cor do texto
    fontDialog1.Font = Texto.Font;
    fontDialog1.Color = Texto.ForeColor;

    // Permitir que o utilizador selecione a cor do texto
    fontDialog1.ShowColor = true;

    if (fontDialog1.ShowDialog() == DialogResult.OK)
    {
        // Modificar a fonte e a cor da caixa de texto
        Texto.Font = fontDialog1.Font;
        Texto.ForeColor = fontDialog1.Color;
    }
}
```

GRUPO IV

Modificações ao Texto

Neste grupo serão implementadas funcionalidades para gerir eventuais modificações que o utilizador faça ao texto do documento. Nomeadamente:

- verificar continuamente que o documento foi modificado
- se o utilizador criar um novo documento (ou abrir um documento) e o documento atual tiver sido modificado, pedir para gravar primeiro
- ao encerrar a aplicação, verificar que o documento foi gravado desde a última vez que foi modificado

34. Para começar, acrescente as seguintes propriedades ao início do ficheiro:

```
/// <summary>
/// Indica se um ficheiro foi modificado desde a última vez que foi gravado.
/// </summary>
private bool textoModificado = false;

/// <summary>
/// Indica se o ficheiro é um novo ficheiro. Esta variável é necessária quando é
aberto um novo ficheiro.
/// </summary>
private bool novoFicheiro = false;
```

35. Acrescente o seguinte código ao evento `MenuFormNovo_Click()` (item `Novo` do menu `Ficheiro`) para:

- Verificar se o ficheiro foi modificado e ainda não foi gravado
- Em caso afirmativo, perguntar ao utilizador o que pretende fazer: gravar o ficheiro, não gravar o ficheiro ou cancelar
 - Se o utilizador responder afirmativamente: gravar o ficheiro
 - Se o utilizador decidir cancelar: terminar esta operação

```
/// <summary>
/// Criar um novo ficheiro.
/// </summary>
private void MenuFormNovo_Click(object sender, EventArgs e)
{
    // Se o ficheiro foi modificado e não gravado, informar o utilizador
    if (textoModificado)
    {
        DialogResult resultado = MessageBox.Show("O ficheiro foi modificado. Deseja gravar?", "Notepad", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

        if (resultado == DialogResult.Yes)
        {
            // Guardar o ficheiro
            Guardar();
        }
        else if (resultado == DialogResult.Cancel)
        {
            // Cancelar a operação de criação de novo ficheiro
            return;
        }
    }

    // Eliminar o conteúdo da caixa de texto
    Texto.Text = string.Empty;

    // Limpar as informações do ficheiro
    textoModificado = false;
    nomeFicheiro = string.Empty;
    localFicheiro = string.Empty;

    // Inicialmente o novo ficheiro não tem nome
    this.Text = "Sem título - Notepad";
}
```

36. Da mesma forma, acrescente o seguinte código ao evento `MenuFormAbrir_Click()` (item `Abrir` do menu `Ficheiro`) para:

- Verificar se o ficheiro foi modificado e ainda não foi gravado

- Em caso afirmativo, perguntar ao utilizador o que pretende fazer: gravar o ficheiro, não gravar o ficheiro ou cancelar
 - Se o utilizador responder afirmativamente: gravar o ficheiro
 - Se o utilizador decidir cancelar: terminar esta operação

```

/// <summary>
/// Abrir um novo ficheiro.
/// </summary>
private void MenuFormAbrir_Click(object sender, EventArgs e)
{
    // Se o ficheiro foi modificado e não gravado, informar o utilizador
    if (textoModificado)
    {
        DialogResult resultado = MessageBox.Show("O ficheiro foi modificado. Deseja gravar?", "Notepad", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

        if (resultado == DialogResult.Yes)
        {
            // Guardar o ficheiro
            Guardar();
        }
        else if (resultado == DialogResult.Cancel)
        {
            // Cancelar a operação de abertura de novo ficheiro
            return;
        }
    }

    // Título
    openFileDialog1.Title = "Abrir";
    // Permitir selecionar apenas um ficheiro
    openFileDialog1.Multiselect = false;
    // Tipos de ficheiros que podem ser abertos
    openFileDialog1.Filter = "Ficheiros de texto (*.txt)|*.txt|Todos os ficheiros (*.*)|*.*";
    openFileDialog1.FilterIndex = 0;

    // Verificar que o nome e a localização especificadas pelo utilizador existem
    openFileDialog1.CheckFileExists = true;
    openFileDialog1.CheckPathExists = true;

    openFileDialog1.FileName = string.Empty;

    // Mostrar o OpenFileDialog e verificar o resultado da operação
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Armazenar o nome e a localização do novo ficheiro
        nomeFicheiro = Path.GetFileName(openFileDialog1.FileName);
        localFicheiro = Path.GetDirectoryName(openFileDialog1.FileName);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";

        // Indicar que o ficheiro ainda não foi modificado
        textoModificado = false;

        // Indicar que foi aberto um novo ficheiro
        novoFicheiro = true;

        // Abrir o ficheiro e colocar o conteúdo na caixa de texto
        Texto.Text = File.ReadAllText(openFileDialog1.FileName, Encoding.UTF8);
    }
}

```

37. Acrescente aos métodos `Guardar()` e `GuardarComo()` código para assinalar que o ficheiro ainda não modificado desde a última gravação:

```

/// <summary>
/// Guardar o ficheiro atual.
/// </summary>
private void Guardar()
{
    // Se o ficheiro ainda não foi guardado, executar o método GuardarComo()
    if ((nomeFicheiro == string.Empty) || (localFicheiro == string.Empty))
    {
        GuardarComo();
    }
    else
    {
        // Guardar o ficheiro
        File.WriteAllText(Path.Combine(localFicheiro, nomeFicheiro), Texto.Text,
        Encoding.UTF8);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";

        // Indicar que não ocorreu modificação desde a última gravação
        textoModificado = false;
    }
}

/// <summary>
/// Guardar o ficheiro atual especificando um nome e localização.
/// </summary>
private void GuardarComo()
{
    saveFileDialog1.Title = "Guardar como";
    saveFileDialog1.Filter = "Ficheiros de texto (*.txt)|*.txt|Todos os ficheiros (*.*)|*.*";
    saveFileDialog1.FilterIndex = 0;
    saveFileDialog1.DefaultExt = "txt";

    // Verificar que a localização especificada pelo utilizador existe
    saveFileDialog1.CheckPathExists = true;

    // Ir para a diretoria do ficheiro (se existir uma localização de ficheiro)
    if (localFicheiro != string.Empty)
    {
        saveFileDialog1.InitialDirectory = localFicheiro;
    }

    // Especificar o nome do ficheiro
    saveFileDialog1.FileName = nomeFicheiro;

    // Mostrar o SaveFileDialog e verificar o resultado da operação
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Armazenar o nome e a localização na qual o ficheiro foi gravado
        nomeFicheiro = Path.GetFileName(saveFileDialog1.FileName);
        localFicheiro = Path.GetDirectoryName(saveFileDialog1.FileName);

        // Mudar o título da janela para o nome do ficheiro
        this.Text = nomeFicheiro + " - Notepad";

        // Gravar o conteúdo do ficheiro
        File.WriteAllText(saveFileDialog1.FileName, Texto.Text, Encoding.UTF8);

        // Indicar que não ocorreu modificação desde a última gravação

```

```

        textoModificado = false;
    }
}

```

38. Para determinar se o texto de um documento foi modificado, iremos utilizar o evento `TextChanged()` da caixa de texto `Texto`. Acrescente o seguinte código ao evento:

```

/// <summary>
/// Este evento é executado sempre que o texto é modificado no controlo Texto.
/// </summary>
private void Texto_TextChanged(object sender, EventArgs e)
{
    // Apenas acrescentar o carater '*' ao título da aplicação se não tiver sido
    // criado ou aberto um novo ficheiro
    if (!novoFicheiro)
    {
        // Mudar o título da janela uma única vez (apenas quando ocorre a
        // modificação pela primeira vez)
        if (!textoModificado)
        {
            this.Text = "* " + this.Text;
        }

        // Indicar que o ficheiro foi modificado
        textoModificado = true;
    }
    else
    {
        novoFicheiro = false;
    }
}

```

Para finalizar: quando o utilizador termina a aplicação, esta deverá verificar se o ficheiro foi modificado (e ainda não foi gravado), dando a possibilidade ao utilizador de gravar o ficheiro ou cancelar a operação de encerramento.

39. Acrescente o evento `FormClosing()` à janela principal da aplicação e insira o seguinte código:

```

/// <summary>
/// Antes de encerrar a aplicação, verificar se o ficheiro foi modificado.
/// </summary>
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    // Se o ficheiro foi modificado e não gravado, informar o utilizador
    if (textoModificado)
    {
        DialogResult resultado = MessageBox.Show("O ficheiro foi modificado. Deseja gravar?", "Notepad", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

        if (resultado == DialogResult.Yes)
        {
            // Guardar o ficheiro
            Guardar();
        }
        else if (resultado == DialogResult.Cancel)
        {
            // Cancelar o encerramento da aplicação
            e.Cancel = true;
            return;
        }
    }
}

```

```
e.Cancel = false;  
}
```

O evento `FormClosing()`

Ocorre quando uma janela (*form*) está prestes a ser fechada.

Utilizando este evento, o programador pode efetuar operações de finalização da aplicação e pode, inclusivé, cancelar o encerramento da mesma.

FIM