

ROTEIRO DE AULA

Tema: Aula 7 – Manipulação de arquivos Word

Data:

Semana:

Tempo estimado: 3 hrs + 3 hrs de atividade

Requisitos e ferramentas para aula:

Visual Studio Code

Gerenciador de pacotes NuGet

Biblioteca Spire.Doc

Descrever roteiro de aula:

- Mostrar [biblioteca Spire](#), suas opções e seu propósito. Clicar em *Tutorials* em Spire.Doc

The screenshot shows the Spire libraries website with a navigation bar for .NET and Java. The main content area displays six product cards, each for a different library. Each card includes a description, a 'Download Free Trial' button, and links for 'Learn More' and 'Tutorials'. The 'Tutorials' link for Spire.Doc is highlighted with a purple box.

Library	Description	Download Free Trial	Learn More	Tutorials
Spire.Doc	A professional Word .NET library designed to create, read, write, convert and print Word document files in any .NET (C#, VB.NET, ASP.NET, .NET Core) application with fast and high quality performance.	Download Free Trial	Learn More	Tutorials
Spire.XLS	A professional Excel .NET library that can be used to create, read, write, convert and print Excel files in any type of .NET (C#, VB.NET, ASP.NET, .NET Core) application.	Download Free Trial	Learn More	Tutorials
Spire.PDF	A professional PDF library applied to creating, writing, editing, handling and reading PDF files without any external dependencies within .NET (C#, VB.NET, ASP.NET, .NET Core) application.	Download Free Trial	Learn More	Tutorials
Spire.Presentation	A professional PowerPoint® compatible library that enables developers to create, read, write, modify, convert and Print PowerPoint documents on any .NET framework, .NET Core platform.	Download Free Trial	Learn More	Tutorials
Spire.Email	A professional .NET Email library specially designed for developers to create, read and manipulate emails from any .NET (C#, VB.NET, ASP.NET) platform with fast and high quality performance.	Download Free Trial	Learn More	Tutorials
Spire.Barcode	A professional barcode library specially designed for .NET developers (C#, VB.NET, ASP.NET, .NET Core) to generate, read and scan 1D & 2D barcodes.	Download Free Trial	Learn More	Tutorials

- Mostrar alguns comandos pela documentação. Clicar em *Create, Write and Save Word in C#, VB.NET*

Spire.Doc

Program Guide for .NET

Conversion

Document Operation

Mail Merge

FormField

Security

Paragraph

Page Setup

Page Background

Text

Image and Shape

Hyperlink

Comment

Table

Bookmark

Header and Footer

Footnote

Spire.Doc Program Guide Content for .NET



Spire.Doc, specially designed for developers/programmers to manipulate word proceeding tasks, from Word version 97–2003 to 2010 and 2013 in .NET, Silverlight and WPF Platform, always welcomes any kind of evaluation or test from either organizations or individuals with no charges.

This professional word library enables users to perform a large range of tasks on Word, as document operation (create, open, edit and save), mail merge, security, format (font, paragraph and page settings), objects (text, image, hyperlink, comment, table, bookmark, header/footer, footnote/endnote etc.). The conversion feature stands out of the peers, which enables developers to realize conversion between most popular formats with high fidelity, including Word to PDF, HTML, RTF, XML, Image formats, TEXT and RTF, XML and HTML files also can be converted to Word back. The following sections have been well-organized to show you how these functions work in detail.

Spire.Doc for Java

Download

Buy Now

Request free customized demo just for you. NEW

Conversion

- Convert Word Doc to XML
- Convert XML to PDF in C#, VB.NET
- Convert Word to PDF
- Convert Word to Text
- Convert Word to HTML in C#, VB.NET
- Convert HTML to Word
- Convert XML to Word

Document Operation

- Create, Write and Save Word in C#, VB.NET
- Solutions to Open Word in C#, VB.NET
- Edit Word Document in C#, VB.NET
- Set Word View Modes in C#, VB.NET
- Merge Multiple Word Documents into One in C#, VB.NET
- Set Word Document Properties

- Comando para criar o documento

Create Word

Spire.Doc for .NET offers a **Document** class to enable developers to initialize a new Document instance. This instance is a new blank Word document.

[C#]

```
//Create New Word
Document doc = new Document();
```

- Comandos para escrita no documento

Write Contents in Word

Generally speaking, contents are written in a paragraph which is contained in section of document. Therefore, you need to add a new section firstly by invoking method **Document.AddSection()**. And add a new paragraph in new section by invoking **Section.AddParagraph()**. After that, you can write contents in paragraph by invoking **Paragraph.AppendText(string text)** method.

[C#]

```
//Add Section
Section section = doc.AddSection();
//Add Paragraph
Paragraph Para = section.AddParagraph();
//Append Text
Para.AppendText("Spire.Doc for .NET, a professional .NET Word component, "
+"enables developers to perform a large range of tasks on Word document(from Version Word97-2003 to Word
2010) "
+"for .NET in C# and VB.NET."
+"This libray is specially designed for .NET developers to help them"
+"to create any WinForm and ASP.NET Web applications to create, open, write, edit, save and convert"
+"Word document without Microsoft Office and any other third-party tools installed on system.");
```

- Comando para salvar o documento

Save Word

Invoke **SaveToFile** method of Document class to save the written Word. The parameter passed to this method is **string fileName**. If you want to set extension as .doc or .docx, you need to pass the other parameter **FileFormat fileFormat** to this method.

[C#]

```
//Save Word
doc.SaveToFile("OperateWord.docx", FileFormat.Docx);
```

[VB.NET]

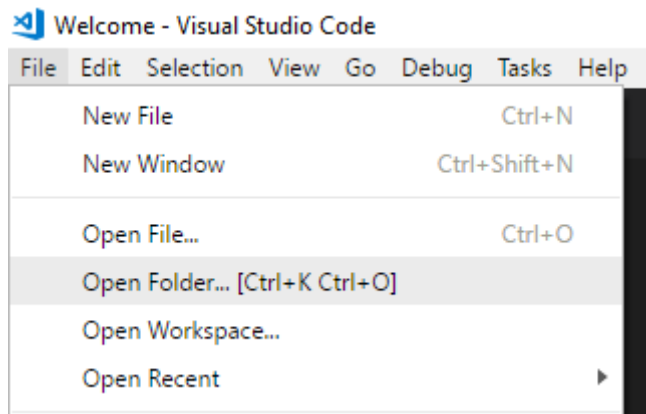
```
'Save Word
doc.SaveToFile("OperateWord.docx", FileFormat.Docx)
```

Spire.Doc, professional Word component, is specially designed for developers to fast generate, write, modify and save Word documents in .NET, Silverlight and WPF with C# and VB.NET. Also, it supports conversion between Word and other popular formats, such as PDF, HTML, Image, Text and so on, in .NET and WPF platform.

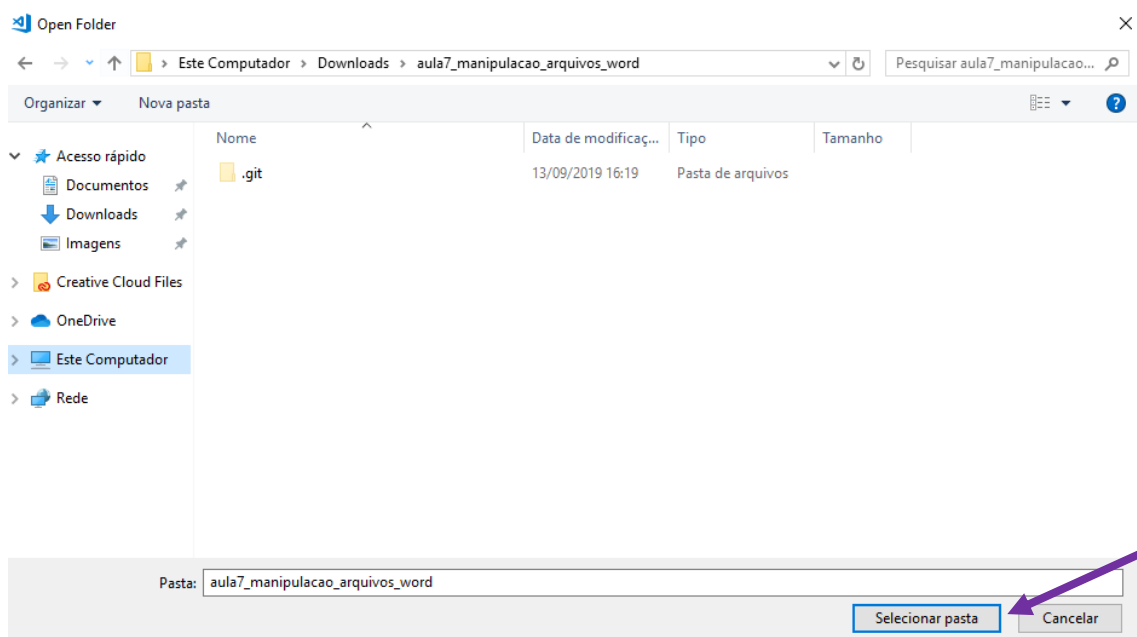
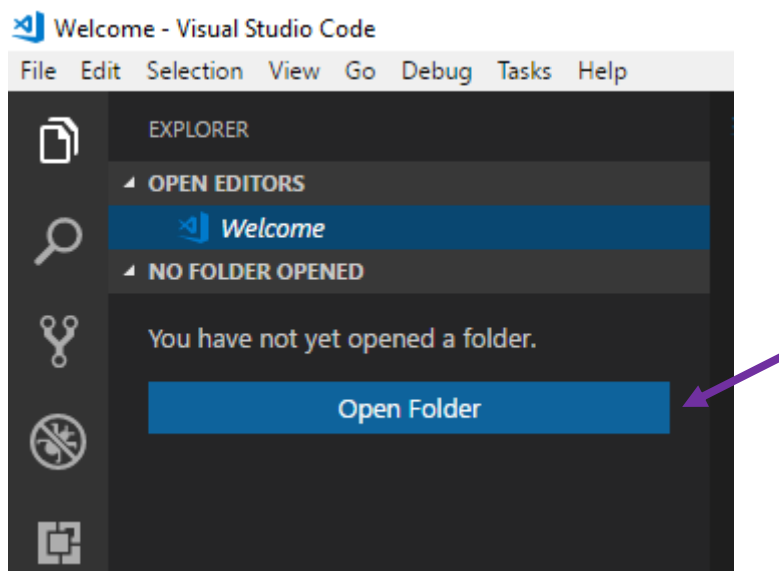
- Criar repositório no GitHub e clonar localmente

The screenshot shows a GitHub repository page for 'aula7_manipulacao_arquivos_word' by user 'Saulomsantos'. At the top, there are buttons for 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A purple arrow points to the 'Clone or download' button. Below this, a modal window titled 'Clone with HTTPS' is open, showing the repository URL: 'https://github.com/Saulomsantos/aula7_'. The repository description is 'Aula 07 do curso CodeXP sobre manipulação de arquivos Word utilizando a biblioteca Spire.Doc'.

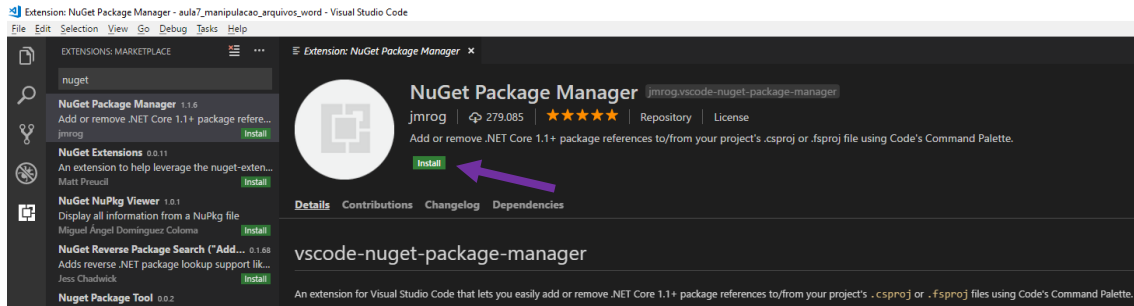
- Abrir a pasta clonada no Visual Studio Code



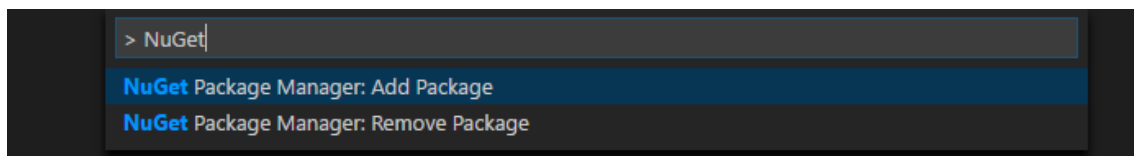
ou



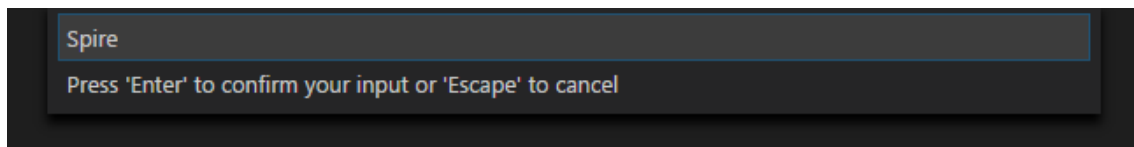
- Abrir o terminal do Visual Studio Code (atalho ctrl + `)
- Criar um novo projeto de console, executando o comando *dotnet new console*
- Instalar a extensão NuGet Package Manager e clicar em reload



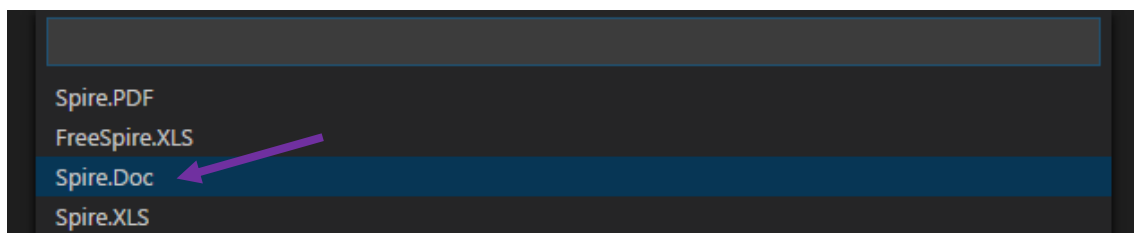
- Executar o comando > NuGet Package Manager : Add Package na barra de comandos do Visual Studio Code (atalho ctrl + p)



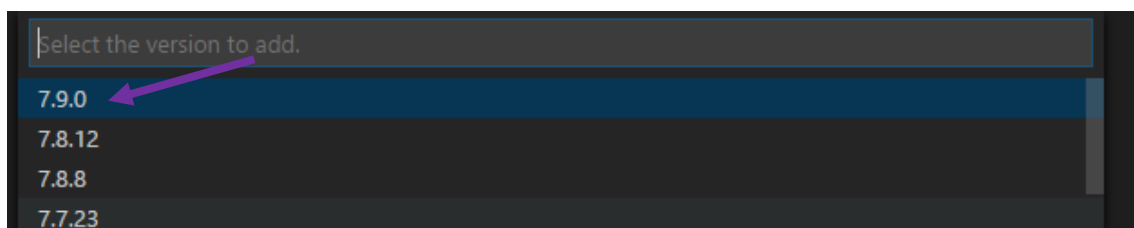
- Digitar a(s) palavra(s) chave(s) no campo de busca e apertar enter



- Selecionar a biblioteca desejada



- Selecionar a versão mais recente



- Criar uma nova pasta *exemplo*
- Dentro da pasta exemplo, criar quatro novas pastas: *doc*, *html*, *img* e *pdf*
- Copiar o arquivo *logo_csharp.png* para a pasta *img*
- No arquivo *Program.cs*, definir que a biblioteca *Spire.Doc* está sendo utilizada para que seja possível usar seus métodos

```
Program.cs x
1 using System;
2 using Spire.Doc;
3
```

- Criar um novo documento através do comando

```
0 references
static void Main(string[] args)
{
    // Cria um documento com o nome exemploDoc
    Document exemploDoc = new Document();|
```

- Em seguida, adicionar uma seção ao documento

```
// Cria uma seção com o nome secaoCapa e adiciona ao documento
// Cada seção pode ser entendida como uma página do arquivo
// Portanto, caso seja adicionada uma nova seção, é feita uma quebra de página no arquivo e uma nova página é criada
Section secaoCapa = exemploDoc.AddSection();
```

- Para que parágrafos possam ser adicionados, é preciso declarar o uso da dependência *Spire.Doc.Documents*

```
Program.cs x
1 using System;
2 using Spire.Doc;
3 using Spire.Doc.Documents;
4
```

- Adicionar um parágrafo para que possa receber conteúdo

```
// Cria um parágrafo com o nome titulo e adiciona à seção secaoCapa
// Os parágrafos são necessários para a inserção de textos, imagens, tabelas etc, ainda que o conteúdo ocupe apenas uma linha
Paragraph titulo = secaoCapa.AddParagraph();
```

- Adicionar um texto ao parágrafo

```
// Adiciona um texto ao parágrafo titulo criado anteriormente
// Assim como em console, é possível utilizar o comando \n para definir uma nova linha
// Ou seja, no comando abaixo é atribuído um texto e dá-se um espaçamento de duas linhas em seguida
titulo.AppendText("Exemplo de titulo\n\n");
```

- Alinhar o parágrafo ao centro horizontalmente

```
// Através da propriedade HorizontalAlignment, é possível alinhar o parágrafo horizontalmente
// No exemplo abaixo, o parágrafo é alinhado ao centro
titulo.Format.HorizontalAlignment = HorizontalAlignment.Center;
```

- Criar um estilo de formatação

```
// Cria um estilo de formatação
// Para definir parâmetros de estilização de parágrafo, cria-se um estilo de formatação e adiciona ao documento
ParagraphStyle estilo01 = new ParagraphStyle(exemploDoc);
```

- É possível definir um nome ao estilo

```
// Define um nome para o estilo
estilo01.Name = "Cor do título";
```

- Para que a cor do texto possa ser definida, é preciso declarar o uso da dependência System.Drawing

```
using System.Drawing;
using Spire.Doc;
using Spire.Doc.Documents;
```

- Definir a cor que será atribuída ao texto

```
// Define a cor do texto
estilo01.CharacterFormat.TextColor = Color.DarkBlue;
```

- Definir que o texto deste estilo será em negrito

```
// Define que o texto deste estilo será em negrito (bold)
estilo01.CharacterFormat.Bold = true;
```

- Adicionar o estilo ao documento

```
// Adiciona o estilo ao documento
exemploDoc.Styles.Add(estilo01);
```

- Aplicar o estilo ao parágrafo

```
// Aplica o estilo ao parágrafo
// O método ApplyStyle recebe como parâmetro o nome do estilo, que pode ser definido pela propriedade (exemplo abaixo)
// ou passando o nome do estilo como string ex: ("Cor do título")
titulo.ApplyStyle(estilo01.Name);
```

- Adicionar um novo parágrafo na mesma seção

```
// Adiciona um novo parágrafo na mesma seção
Paragraph capa = secaoCapa.AddParagraph();
```

- Adicionar um texto a este parágrafo com tabulação

```
// Adiciona um texto ao parágrafo com tabulação (\t é semelhante à tecla tab)
textoCapa.AppendText("\tEste é um exemplo de criação de um parágrafo utilizando a biblioteca Spire.Doc\n");
```

- Adicionar mais um parágrafo na mesma seção

```
// Adiciona mais um novo parágrafo na mesma seção
Paragraph textoCapa2 = secaoCapa.AddParagraph();
```

- Adicionar um texto a este parágrafo com concatenação

```
// Adiciona um texto ao parágrafo com concatenação
textoCapa2.AppendText("\tBasicamente, então, uma seção representa uma página e os parágrafos dentro de uma mesma seção, " +
"obviamente, aparecem na mesma página.");
```

- Adicionar um novo parágrafo na mesma seção e um texto explicativo, e centralizado, a este parágrafo

```
// Adiciona um novo parágrafo na mesma seção
Paragraph imagemCapa = secaoCapa.AddParagraph();

// Adiciona um texto ao parágrafo
imagemCapa.AppendText("\n\n\tAgora vamos inserir uma imagem em um parágrafo\n\n");

// Centraliza o texto acima
imagemCapa.Format.HorizontalAlignment = HorizontalAlignment.Center;
```

- Para se trabalhar com imagens, é preciso declarar o uso da dependência Spire.Doc.Fields

```
using Spire.Doc.Documents;
using Spire.Doc.Fields;
```


- Adicionar a imagem ao parágrafo

```
// Adiciona uma imagem ao parágrafo
DocPicture imagemExemplo = imagemCapa.AppendPicture(Image.FromFile(@"exemplo\img\logo_csharp.png"));
```

- Definir largura e altura da imagem

```
// Define a largura e a altura da imagem
imagemExemplo.Width = 300;
imagemExemplo.Height = 300;
```

- Adicionar uma nova seção ao documento com um parágrafo e um texto atribuído a ele

```
// Adiciona uma nova seção
Section secaoCorpo = exemploDoc.AddSection();

// Adiciona um novo parágrafo a esta nova seção
Paragraph paragrafoCorpo1 = secaoCorpo.AddParagraph();

// Adiciona um texto a este parágrafo
paragrafoCorpo1.AppendText("\tEste é um exemplo de criação de um parágrafo em uma nova página, após uma quebra de seção. "
"Assim como quando utilizamos variáveis, é possível fechar aspas, inserir um sinal '+' e continuar o parágrafo.\n\n" +
"\tComo foi criada outra seção, perceba que o parágrafo acima começou em outra página.\n");
```

- Adicionar uma tabela nesta seção

```
// Adiciona uma tabela
Table tabela = secaoCorpo.AddTable(true);
```

- Criar o cabeçalho da tabela

```
// Cria o cabeçalho da tabela
String[] cabecalho = {"Item", "Descrição", "Qtd.", "Preço Unit", "Preço"};
```

- Criar os dados da tabela

```
// Cria os dados da tabela
String[][] dados = {
    new String[]{"Cenoura", "Vegetal muito nutritivo", "1", "R$ 4,00", "R$ 4,00"},
    new String[]{"Batata", "Vegetal muito consumido", "2", "R$ 5,00", "R$ 10,00"},
    new String[]{"Alface", "Vegetal utilizado desde 500 a.C.", "1", "R$ 1,50", "R$ 1,50"},
    new String[]{"Tomate", "Tomate é uma fruta", "2", "R$ 6,00", "R$ 12,00"}
};
```

- Adicionar as células da tabela

```
// Adiciona as células da tabela
tabela.ResetCells(dados.Length + 1, cabecalho.Length);
```

- Adicionar uma linha na posição [0] do array de linhas

```
// Adiciona uma linha na posição [0] do array de linhas e define que ela é o cabeçalho
TableRow Linha1 = tabela.Rows[0];
Linha1.IsHeader = true;
```

- Definir a altura da linha

```
// Define a altura da linha
Linha1.Height = 23;
```

- Formatar o cabeçalho

```
// Formatação do cabeçalho
Linha1.RowFormat.BackColor = Color.AliceBlue;
for (int i = 0; i < cabecalho.Length; i++)
{
    // Alinhamento das células
    Paragraph p = Linha1.Cells[i].AddParagraph();
    Linha1.Cells[i].CellFormat.VerticalAlignment = VerticalAlignment.Middle;
    p.Format.HorizontalAlignment = HorizontalAlignment.Center;

    // Formatação dos dados do cabeçalho
    TextRange TR = p.AppendText(cabecalho[i]);
    TR.CharacterFormat.FontName = "Calibri";
    TR.CharacterFormat.FontSize = 14;
    TR.CharacterFormat.TextColor = Color.Teal;
    TR.CharacterFormat.Bold = true;
}
```

- Adicionar as linhas do corpo da tabela

```
// Adiciona as linhas do corpo da tabela
for (int r = 0; r < dados.Length; r++)
{
    TableRow LinhaDados = tabela.Rows[r + 1];

    // Define a altura da linha
    LinhaDados.Height = 20;

    // Percorre as colunas
    for (int c = 0; c < dados[r].Length; c++)
    {
        // Alinha as células
        LinhaDados.Cells[c].CellFormat.VerticalAlignment = VerticalAlignment.Middle;

        // Preenche os dados nas linhas
        Paragraph p2 = LinhaDados.Cells[c].AddParagraph();
        TextRange TR2 = p2.AppendText(dados[r][c]);

        // Formata as células
        p2.Format.HorizontalAlignment = HorizontalAlignment.Center;
        TR2.CharacterFormat.FontName = "Calibri";
        TR2.CharacterFormat.FontSize = 12;
        TR2.CharacterFormat.TextColor = Color.Brown;
    }
}
```

- Salvar o arquivo em Docx

```
// Salvando o arquivo
// Utiliza o método SaveToFile para salvar com o nome e o formato escolhido
// Assim como no Word, caso já exista um arquivo com o mesmo nome, este será substituído por um novo
exemploDoc.SaveToFile(@"exemplo\exemplo_arquivo_word.docx", FileFormat.Docx);
```

Descrever atividades:

1. Criar um documento .docx utilizando C# e a biblioteca Spire.Doc

2. O documento deve conter:

2.1. Seção 1:

2.1.1. Título centralizado

2.1.2. Um parágrafo com um texto e várias estilizações

2.1.3. Uma tabela com quatro colunas e três linhas

2.2. Seção 2:

2.2.1. Uma imagem centralizada

2.2.2. Um parágrafo centralizado com a descrição da imagem

2.2.3. Exportar para .pdf

Tempo atividades: 3 horas

Descrever atividade extra:

3. Desafios:

3.1. Exportar o exercício para uma página .html

3.2. Adicionar um parágrafo com um texto e estilos diferentes na mesma frase, em cada trecho desta