



EBook Gratis

APRENDIZAJE epplus

Free unaffiliated eBook created from
Stack Overflow contributors.

#epplus

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con eplus.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Instalación.....	3
Empezando.....	3
Capítulo 2: Adjuntar datos al documento existente.....	5
Introducción.....	5
Examples.....	5
Anexando datos.....	5
Capítulo 3: Columnas y filas.....	6
Introducción.....	6
Examples.....	6
Columnas de autoajuste.....	6
Ocultar columnas y filas.....	6
Redimensionar filas y columnas.....	6
Copiar columnas o filas.....	7
Capítulo 4: Combinar células.....	8
Introducción.....	8
Examples.....	8
Fusionando celulas.....	8
Capítulo 5: Creando cuadros.....	9
Introducción.....	9
Examples.....	9
Gráfico circular.....	9
Gráfico de linea.....	9
Capítulo 6: Creando fórmulas y calculando rangos.....	11
Introducción.....	11
Examples.....	11

Añadir fórmulas a una celda.....	11
Fórmula con múltiples hojas.....	11
Calculo manual.....	11
Ejemplo completo con fórmulas.....	12
Capítulo 7: Diseño del documento de Excel.....	13
Introducción.....	13
Examples.....	13
Color de fondo.....	13
Estilos de borde.....	13
Estilos de fuente.....	13
Alineación de texto y ajuste de palabras.....	14
Ejemplo completo con todos los estilos.....	14
Añadir una imagen a una hoja.....	15
Capítulo 8: Formato de valores.....	16
Introducción.....	16
Examples.....	16
Formateo de numero.....	16
Formato de fecha.....	16
Formato de texto.....	16
Capítulo 9: Guardando el documento de Excel.....	18
Introducción.....	18
Examples.....	18
Guardar en el disco.....	18
Enviar al navegador.....	19
Guardar en disco con SaveFileDialog.....	19
Capítulo 10: Importando datos desde archivo existente.....	21
Introducción.....	21
Examples.....	21
Importar datos desde archivo Excel.....	21
Importar datos desde archivo CSV.....	21
Importar datos desde un archivo de Excel con FileUpload Control.....	22
Crear un DataTable desde un archivo de Excel.....	22

Capítulo 11: Mesas	24
Introducción	24
Examples	24
Añadiendo y formando una tabla	24
Capítulo 12: Rellenando el documento con datos	25
Introducción	25
Examples	25
Rellenar con una tabla de datos	25
Rellene con una tabla de datos desde una consulta SQL o un procedimiento almacenado	25
Rellenar manualmente las celdas	26
Rellenar desde colección	26
Capítulo 13: Tabla dinámica	28
Introducción	28
Examples	28
Creación de una tabla dinámica	28
Capítulo 14: Texto enriquecido en celdas	30
Introducción	30
Examples	30
Añadiendo RichText a una celda	30
Propiedades de formato de texto	31
Insertando RichText en una celda	31
Capítulo 15: Validación de entrada de usuario	33
Introducción	33
Examples	33
Validación de lista	33
Validación de enteros	33
Validación de fecha y hora	34
Validación de la longitud del texto	34
Creditos	35

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [epplus](#)

It is an unofficial and free epplus ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official epplus.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con epplus

Observaciones

EPPlus es una biblioteca .NET que lee y escribe archivos de Excel 2007/2010/2013 utilizando el formato Open Office Xml (xlsx).

EPPlus soporta:

- Rangos celulares
- Estilo de celda (borde, color, relleno, fuente, número, alineaciones)
- Cartas
- Imágenes
- Formas
- Comentarios
- Mesas
- Protección
- Cifrado
- Tablas dinámicas
- Validación de datos
- Formato condicional
- VBA
- Cálculo de fórmula

Versiones

Versión	Fecha de lanzamiento
Primer lanzamiento	2009-11-30
2.5.0.1	2010-01-25
2.6.0.1	2010-03-23
2.7.0.1	2010-06-17
2.8.0.2	2010-11-15
2.9.0.1	2011-05-31
3.0.0.2	2012-01-31
3.1	2012-04-11
4.0.5	2016-01-08
4.1	2016-07-14

Examples

Instalación

Descargue los archivos de [CodePlex](#) y agréguelos al proyecto.

O instale los archivos con el Administrador de paquetes.

```
PM> Install-Package EPPlus
```

Empezando

```
//Create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //Set some properties of the Excel document
    excelPackage.Workbook.Properties.Author = "VDWWD";
    excelPackage.Workbook.Properties.Title = "Title of Document";
    excelPackage.Workbook.Properties.Subject = "EPPlus demo export data";
    excelPackage.Workbook.Properties.Created = DateTime.Now;

    //Create the WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //Add some text to cell A1
    worksheet.Cells["A1"].Value = "My first EPPlus spreadsheet!";
    //You could also use [line, column] notation:
    worksheet.Cells[1,2].Value = "This is cell B1!";

    //Save your file
    FileInfo fi = new FileInfo(@"Path\To\Your\File.xlsx");
    excelPackage.SaveAs(fi);
}

//Opening an existing Excel file
FileInfo fi = new FileInfo(@"Path\To\Your\File.xlsx");
using (ExcelPackage excelPackage = new ExcelPackage(fi))
{
    //Get a WorkSheet by index. Note that EPPlus indexes are base 1, not base 0!
    ExcelWorksheet firstWorksheet = excelPackage.Workbook.Worksheets[1];

    //Get a WorkSheet by name. If the worksheet doesn't exist, throw an exeption
    ExcelWorksheet namedWorksheet = excelPackage.Workbook.Worksheets["SomeWorksheet"];

    //If you don't know if a worksheet exists, you could use LINQ,
    //So it doesn't throw an exception, but return null in case it doesn't find it
    ExcelWorksheet anotherWorksheet =
        excelPackage.Workbook.Worksheets.FirstOrDefault(x=>x.Name=="SomeWorksheet");

    //Get the content from cells A1 and B1 as string, in two different notations
    string valA1 = firstWorksheet.Cells["A1"].Value.ToString();
    string valB1 = firstWorksheet.Cells[1,2].Value.ToString();

    //Save your file
    excelPackage.Save();
}
```

Lea Empezando con epplus en línea: <https://riptutorial.com/es/epplus/topic/8070/empezando-con-epplus>

Capítulo 2: Adjuntar datos al documento existente

Introducción

Cómo anexar datos a un documento de Excel ya existente.

Examples

Anexando datos

```
//the path of the file
string filePath = "C:\\ExcelDemo.xlsx";

//or if you use asp.net, get the relative path
filePath = Server.MapPath("ExcelDemo.xlsx");

//create a fileinfo object of an excel file on the disk
FileInfo file = new FileInfo(filePath);

//create a new Excel package from the file
using (ExcelPackage excelPackage = new ExcelPackage(file))
{
    //create an instance of the the first sheet in the loaded file
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets[1];

    //add some data
    worksheet.Cells[4, 1].Value = "Added data in Cell A4";
    worksheet.Cells[4, 2].Value = "Added data in Cell B4";

    //save the changes
    excelPackage.Save();
}
```

Lea Adjuntar datos al documento existente en línea:

<https://riptutorial.com/es/epplus/topic/8596/adjuntar-datos-al-documento-existente>

Capítulo 3: Columnas y filas

Introducción

Este tema contiene información sobre el trabajo con columnas y filas, como cambio de tamaño, ocultación, ajuste automático

Examples

Columnas de autoajuste

```
//Make all text fit the cells
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns();

//Autofit with minimum size for the column.
double minimumSize = 10;
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns(minimumSize);

//Autofit with minimum and maximum size for the column.
double maximumSize = 50;
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns(minimumSize, maximumSize);

//optional use this to make all columns just a bit wider, text would sometimes still overflow
after AutoFitColumns().
for (int col = 1; col <= worksheet.Dimension.End.Column; col++)
{
    worksheet.Column(col).Width = worksheet.Column(col).Width + 1;
}
```

Ocultar columnas y filas

```
//Hide column "A"
worksheet.Column(1).Hidden = true;

//Hide row 1
worksheet.Row(1).Hidden = true;
```

Redimensionar filas y columnas

```
//Set the row "A" height to 15
double rowHeight = 15;
worksheet.Row(1).Height = rowHeight;

//Set the column 1 width to 50
double columnWidth = 50;
worksheet.Column(1).Width = columnWidth;
```

Cuando Bestfit se establece en verdadero, la columna se ampliará cuando un usuario ingrese números en una celda

```
worksheet.Column(1).BestFit = true;
```

Copiar columnas o filas

```
workSheet.Cells[1, 5, 100, 5].Copy(workSheet.Cells[1, 2, 100, 2]);
```

Copia la columna 5 en la columna 2 Básicamente Source.Copy (Destino)

Esto solo copiaría las primeras 100 filas.

```
Cells[RowStart, ColumnStart, RowEnd, ColumnEnd ]  
is the format so to copy a row into another row you would just switch the indexes accordingly
```

Lea Columnas y filas en línea: <https://riptutorial.com/es/epplus/topic/8766/columnas-y-filas>

Capítulo 4: Combinar células

Introducción

Cómo fusionar células

Examples

Fusionando celulas

```
//By range address  
worksheet.Cells["A1:B5"].Merge = true;  
  
//By indexes  
worksheet.Cells[1,1,5,2].Merge = true;
```

Lea Combinar células en línea: <https://riptutorial.com/es/eplus/topic/8728/combinar-celulas>

Capítulo 5: Creando cuadros

Introducción

Cómo crear gráficos con EPPlus

Examples

Gráfico circular

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //fill cell data with a loop, note that row and column indexes start at 1
    Random rnd = new Random();
    for (int i = 1; i <= 10; i++)
    {
        worksheet.Cells[1, i].Value = "Value " + i;
        worksheet.Cells[2, i].Value = rnd.Next(5, 15);
    }

    //create a new piechart of type Pie3D
    ExcelPieChart pieChart = worksheet.Drawings.AddChart("pieChart", eChartType.Pie3D) as
ExcelPieChart;

    //set the title
    pieChart.Title.Text = "PieChart Example";

    //select the ranges for the pie. First the values, then the header range
    pieChart.Series.Add(ExcelRange.GetAddress(2, 1, 2, 10), ExcelRange.GetAddress(1, 1, 1,
10));

    //position of the legend
    pieChart.Legend.Position = eLegendPosition.Bottom;

    //show the percentages in the pie
    pieChart.DataLabel.ShowPercent = true;

    //size of the chart
    pieChart.SetSize(500, 400);

    //add the chart at cell C5
    pieChart.SetPosition(4, 0, 2, 0);
}
```

Gráfico de linea

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
```

```

//create a WorkSheet
ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

//fill cell data with a loop, note that row and column indexes start at 1
Random rnd = new Random();
for (int i = 2; i <= 11; i++)
{
    worksheet.Cells[1, i].Value = "Value " + (i - 1);
    worksheet.Cells[2, i].Value = rnd.Next(5, 25);
    worksheet.Cells[3, i].Value = rnd.Next(5, 25);
}
worksheet.Cells[2, 1].Value = "Age 1";
worksheet.Cells[3, 1].Value = "Age 2";

//create a new piechart of type Line
ExcelLineChart lineChart = worksheet.Drawings.AddChart("lineChart", eChartType.Line) as
ExcelLineChart;

//set the title
lineChart.Title.Text = "LineChart Example";

//create the ranges for the chart
var rangeLabel = worksheet.Cells["B1:K1"];
var range1 = worksheet.Cells["B2:K2"];
var range2 = worksheet.Cells["B3:K3"];

//add the ranges to the chart
lineChart.Series.Add(range1, rangeLabel);
lineChart.Series.Add(range2, rangeLabel);

//set the names of the legend
lineChart.Series[0].Header = worksheet.Cells["A2"].Value.ToString();
lineChart.Series[1].Header = worksheet.Cells["A3"].Value.ToString();

//position of the legend
lineChart.Legend.Position = eLegendPosition.Right;

//size of the chart
lineChart.SetSize(600, 300);

//add the chart at cell B6
lineChart.SetPosition(5, 0, 1, 0);
}

```

Lea Creando cuadros en línea: <https://riptutorial.com/es/epplus/topic/8286/creando-cuadros>

Capítulo 6: Creando fórmulas y calculando rangos.

Introducción

Ejemplos básicos de cómo crear celdas con una fórmula para cálculos dentro de la hoja de Excel.

Examples

Añadir fórmulas a una celda

```
//set the total value of cells in range A1 - A25 into A27
worksheet.Cells["A27"].Formula = "=SUM(A1:A25)";

//set the number of cells with content in range C1 - C25 into C27
worksheet.Cells["C27"].Formula = "=COUNT(C1:C25)";

//fill column K with the sum of each row, range A - J
for (int i = 1; i <= 25; i++)
{
    var cell = worksheet.Cells[i, 12];
    cell.Formula = "=SUM(" + worksheet.Cells[i, 1].Address + ":" + worksheet.Cells[i,
10].Address + ")";
}

//calculate the quartile of range E1 - E25 into E27
worksheet.Cells[27, 5].Formula = "=QUARTILE(E1:E25,1)";
```

Fórmula con múltiples hojas

```
//set the total value of all cells in Sheet 2 into G27
worksheet.Cells["G27"].Formula = "=SUM('" + worksheet2.Name + "'!" +
worksheet2.Dimension.Start.Address + ":" + worksheet2.Dimension.End.Address + ")";

//set the number of cells with content in Sheet 2, range C1 - C25 into I27
worksheet.Cells["I27"].Formula = "=COUNT('" + excelPackage.Workbook.Worksheets[2].Name + "'!" +
+ excelPackage.Workbook.Worksheets[2].Cells["A1:B25"] + ")";
```

Calculo manual

Si usa fórmulas, Excel le pedirá que guarde el archivo cada vez, incluso si no se realizaron cambios. Para evitar este comportamiento, puede establecer el modo de cálculo en manual.

```
excelPackage.Workbook.CalcMode = ExcelCalcMode.Manual;

//fill the sheet with data and set the formulas

excelPackage.Workbook.Calculate();
```

Ejemplo completo con fórmulas

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create 2 WorkSheets
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");
    ExcelWorksheet worksheet2 = excelPackage.Workbook.Worksheets.Add("Sheet 2");

    //set the calculation mode to manual
    excelPackage.Workbook.CalcMode = ExcelCalcMode.Manual;

    //fill cell data with a loop, note that row and column indexes start at 1
    for (int i = 1; i <= 25; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            worksheet.Cells[i, j].Value = (i + j) - 1;
            worksheet2.Cells[i, j].Value = (i + j) - 1;
        }
    }

    //set the total value of cells in range A1 - A25 into A27
    worksheet.Cells["A27"].Formula = "=SUM(A1:A25)";

    //set the number of cells with content in range C1 - C25 into C27
    worksheet.Cells["C27"].Formula = "=COUNT(C1:C25)";

    //fill column K with the sum of each row, range A - J
    for (int i = 1; i <= 25; i++)
    {
        var cell = worksheet.Cells[i, 12];
        cell.Formula = "=SUM(" + worksheet.Cells[i, 1].Address + ":" + worksheet.Cells[i,
10].Address + ")";
    }

    //calculate the quartile of range E1 - E25 into E27
    worksheet.Cells[27, 5].Formula = "=QUARTILE(E1:E25,1)";

    //set the total value of all cells in Sheet 2 into G27
    worksheet.Cells["G27"].Formula = "=SUM('" + worksheet2.Name + "'!" +
worksheet2.Dimension.Start.Address + ":" + worksheet2.Dimension.End.Address + ")";

    //set the number of cells with content in Sheet 2, range C1 - C25 into I27
    worksheet.Cells["I27"].Formula = "=COUNT('" + excelPackage.Workbook.Worksheets[2].Name +
"'!" + excelPackage.Workbook.Worksheets[2].Cells["A1:B25"] + ")";

    //calculate all the values of the formulas in the Excel file
    excelPackage.Workbook.Calculate();

    //Save the file
    FileInfo fi = new FileInfo("FormulaExample.xlsx");
    excelPackage.SaveAs(fi);
}
```

Lea **Creando fórmulas y calculando rangos. en línea:**

<https://riptutorial.com/es/epplus/topic/8227/creando-formulas-y-calculando-rangos->

Capítulo 7: Diseño del documento de Excel

Introducción

Cómo diseñar celdas con tipos de fuente, color de fondo, estilos de borde, etc.

Examples

Color de fondo

```
//fill column A with solid red color from hex
worksheet.Column(1).Style.Fill.PatternType = ExcelFillStyle.Solid;
worksheet.Column(1).Style.Fill.BackgroundColor.SetColor(ColorTranslator.FromHtml("#FF0000"));

//fill row 4 with striped orange background
worksheet.Row(4).Style.Fill.PatternType = ExcelFillStyle.DarkHorizontal;
worksheet.Row(4).Style.Fill.BackgroundColor.SetColor(Color.Orange);
```

Estilos de borde

```
//make the borders of cell F6 thick
worksheet.Cells[6, 6].Style.Border.Top.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Right.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Bottom.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Left.Style = ExcelBorderStyle.Thick;

//make the borders of cells A18 - J18 double and with a purple color
worksheet.Cells["A18:J18"].Style.Border.Top.Style = ExcelBorderStyle.Double;
worksheet.Cells["A18:J18"].Style.Border.Bottom.Style = ExcelBorderStyle.Double;
worksheet.Cells["A18:J18"].Style.Border.Top.Color.SetColor(Color.Purple);
worksheet.Cells["A18:J18"].Style.Border.Bottom.Color.SetColor(Color.Purple);
```

Estilos de fuente

```
//set the font type for cells C1 - C30
worksheet.Cells["C1:C30"].Style.Font.Size = 13;
worksheet.Cells["C1:C30"].Style.Font.Name = "Calibri";
worksheet.Cells["C1:C30"].Style.Font.Bold = true;
worksheet.Cells["C1:C30"].Style.Font.Color.SetColor(Color.Blue);

//Multiple Fonts in the same cell
ExcelRange rg = worksheet.Cells["A1"];
rg.IsRichText = true;
//ExcelRichText uses "using OfficeOpenXml.Style;"
ExcelRichText text1 = rg.RichText.Add("Text with Font1");
text1.Bold = true;
text1.Italic = true;
text1.Color = System.Drawing.Color.Blue;
ExcelRichText text2 = rg.RichText.Add("Text with Font2");
text2.UnderLine = true;
text2.Bold = false;
```

```
text2.Color = System.Drawing.Color.Red;
ExcelRichText text3 = rg.RichText.Add("Text with Font3");
text3.UnderLine = false;
text3.Strike = true;
```

Alineación de texto y ajuste de palabras

```
//make column H wider and set the text align to the top and right
worksheet.Column(8).Width = 25;
worksheet.Column(8).Style.HorizontalAlignment = ExcelHorizontalAlignment.Right;
worksheet.Column(8).Style.VerticalAlignment = ExcelVerticalAlignment.Top;

//wrap text in the cells
worksheet.Column(8).Style.WrapText = true;
```

Ejemplo completo con todos los estilos.

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create the WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some dummy data, note that row and column indexes start at 1
    for (int i = 1; i <= 30; i++)
    {
        for (int j = 1; j <= 15; j++)
        {
            worksheet.Cells[i, j].Value = "Row " + i + ", Column " + j;
        }
    }

    //fill column A with solid red color
    worksheet.Column(1).Style.Fill.PatternType = ExcelFillStyle.Solid;

    worksheet.Column(1).Style.Fill.BackgroundColor.SetColor(ColorTranslator.FromHtml("#FF0000"));

    //set the font type for cells C1 - C30
    worksheet.Cells["C1:C30"].Style.Font.Size = 13;
    worksheet.Cells["C1:C30"].Style.Font.Name = "Calibri";
    worksheet.Cells["C1:C30"].Style.Font.Bold = true;
    worksheet.Cells["C1:C30"].Style.Font.Color.SetColor(Color.Blue);

    //fill row 4 with striped orange background
    worksheet.Row(4).Style.Fill.PatternType = ExcelFillStyle.DarkHorizontal;
    worksheet.Row(4).Style.Fill.BackgroundColor.SetColor(Color.Orange);

    //make the borders of cell F6 thick
    worksheet.Cells[6, 6].Style.Border.Top.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Right.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Bottom.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Left.Style = ExcelBorderStyle.Thick;

    //make the borders of cells A18 - J18 double and with a purple color
    worksheet.Cells["A18:J18"].Style.Border.Top.Style = ExcelBorderStyle.Double;
    worksheet.Cells["A18:J18"].Style.Border.Bottom.Style = ExcelBorderStyle.Double;
    worksheet.Cells["A18:J18"].Style.Border.Top.Color.SetColor(Color.Purple);
}
```

```

worksheet.Cells["A18:J18"].Style.Border.Bottom.Color.SetColor(Color.Purple);

//make all text fit the cells
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns();

//i use this to make all columns just a bit wider, text would sometimes still overflow
after AutoFitColumns(). Bug?
for (int col = 1; col <= worksheet.Dimension.End.Column; col++)
{
    worksheet.Column(col).Width = worksheet.Column(col).Width + 1;
}

//make column H wider and set the text align to the top and right
worksheet.Column(8).Width = 25;
worksheet.Column(8).Style.HorizontalAlignment = ExcelHorizontalAlignment.Right;
worksheet.Column(8).Style.VerticalAlignment = ExcelVerticalAlignment.Top;

//get the image from disk
using (System.Drawing.Image image =
System.Drawing.Image.FromFile(HttpContext.Current.Server.MapPath("logo.jpg")))
{
    var excelImage = worksheet.Drawings.AddPicture("My Logo", image);

    //add the image to row 20, column E
    excelImage.SetPosition(20, 0, 5, 0);
}
}

```

Añadir una imagen a una hoja

```

//get the image from disk
using (System.Drawing.Image image =
System.Drawing.Image.FromFile(HttpContext.Current.Server.MapPath("logo.jpg")))
{
    var excelImage = worksheet.Drawings.AddPicture("My Logo", image);

    //add the image to row 20, column E
    excelImage.SetPosition(20, 0, 5, 0);
}

```

Lea Diseño del documento de Excel en línea: <https://riptutorial.com/es/epplus/topic/8219/disenodel-documento-de-excel>

Capítulo 8: Formato de valores

Introducción

Cómo obtener el formato deseado de los valores DateTime y Numeric.

Examples

Formateo de numero

```
//integer (not really needed unless you need to round numbers, Excel with use default cell
properties)
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0";

//integer without displaying the number 0 in the cell
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "#";

//number with 1 decimal place
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0.0";

//number with 2 decimal places
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0.00";

//number with 2 decimal places and thousand separator
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "#,##0.00";

//number with 2 decimal places and thousand separator and money symbol
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "€#,##0.00";

//percentage (1 = 100%, 0.01 = 1%)
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0%";
```

Formato de fecha

```
//default DateTime patterns
worksheet.Cells["A1:A25"].Style.Numberformat.Format =
DateTimeFormatInfo.CurrentInfo.ShortDatePattern;

//custom DateTime patters
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "dd-MM-yyyy HH:mm";

//or overwrite the patterns in the CurrentThread with your own
Thread.CurrentThread.CurrentCulture = new CultureInfo("nl-NL")
{
    DateTimeFormat = { YearMonthPattern = "MMM yy" }
};
worksheet.Cells["A1:A25"].Style.Numberformat.Format =
DateTimeFormatInfo.CurrentInfo.YearMonthPattern;
```

Formato de texto

```
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "@";
```

Lea Formato de valores en línea: <https://riptutorial.com/es/eplusplus/topic/8080/formato-de-valores>

Capítulo 9: Guardando el documento de Excel

Introducción

Ejemplos sobre cómo guardar la hoja de Excel creada en el disco o enviarla al navegador.

Examples

Guardar en el disco

```
//Using File.WriteAllBytes
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";

    //convert the excel package to a byte array
    byte[] bin = excelPackage.GetAsByteArray();

    //the path of the file
    string filePath = "C:\\ExcelDemo.xlsx";

    //or if you use asp.net, get the relative path
    filePath = Server.MapPath("ExcelDemo.xlsx");

    //write the file to the disk
    File.WriteAllBytes(filePath, bin);

    //Instead of converting to bytes, you could also use FileInfo
    FileInfo fi = new FileInfo(filePath);
    excelPackage.SaveAs(fi);
}

//Using SaveAs
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";
    //the path of the file
    string filePath = "C:\\ExcelDemo.xlsx";

    //or if you use asp.net, get the relative path
    filePath = Server.MapPath("ExcelDemo.xlsx");

    //Write the file to the disk
    FileInfo fi = new FileInfo(filePath);
```

```
    excelPackage.SaveAs(fi);  
}
```

Enviar al navegador

```
//create a new ExcelPackage  
using (ExcelPackage excelPackage = new ExcelPackage())  
{  
    //create the WorkSheet  
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");  
  
    //add some text to cell A1  
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";  
  
    //convert the excel package to a byte array  
    byte[] bin = excelPackage.GetAsByteArray();  
  
    //clear the buffer stream  
    Response.ClearHeaders();  
    Response.Clear();  
    Response.Buffer = true;  
  
    //set the correct contenttype  
    Response.ContentType = "application/vnd.openxmlformats-  
officedocument.spreadsheetml.sheet";  
  
    //set the correct length of the data being send  
    Response.AddHeader("content-length", bin.Length.ToString());  
  
    //set the filename for the excel package  
    Response.AddHeader("content-disposition", "attachment; filename=\"ExcelDemo.xlsx\"");  
  
    //send the byte array to the browser  
    Response.OutputStream.Write(bin, 0, bin.Length);  
  
    //cleanup  
    Response.Flush();  
    HttpContext.Current.ApplicationInstance.CompleteRequest();  
}
```

Guardar en disco con SaveFileDialog

```
//Using File.WriteAllBytes  
using (ExcelPackage excelPackage = new ExcelPackage())  
{  
    //create a new Worksheet  
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");  
  
    //add some text to cell A1  
    worksheet.Cells["A1"].Value = "My fourth EPPlus spreadsheet!";  
  
    //convert the excel package to a byte array  
    byte[] bin = excelPackage.GetAsByteArray();  
  
    //create a SaveFileDialog instance with some properties  
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();  
    saveFileDialog1.Title = "Save Excel sheet";  
}
```

```

saveFileDialog1.Filter = "Excel files|*.xlsx|All files|*.*";
saveFileDialog1.FileName = "ExcelSheet_" + DateTime.Now.ToString("dd-MM-yyyy") + ".xlsx";

//check if user clicked the save button
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    //write the file to the disk
    File.WriteAllBytes(saveFileDialog1.FileName, bin);
}
}

//Using SaveAs
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My fourth EPPlus spreadsheet!";

    //create a SaveFileDialog instance with some properties
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Title = "Save Excel sheet";
    saveFileDialog1.Filter = "Excel files|*.xlsx|All files|*.*";
    saveFileDialog1.FileName = "ExcelSheet_" + DateTime.Now.ToString("dd-MM-yyyy") + ".xlsx";

    //check if user clicked the save button
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        //Get the FileInfo
        FileInfo fi = new FileInfo(saveFileDialog1.FileName);
        //write the file to the disk
        excelPackage.SaveAs(fi);
    }
}
}

```

Lea Guardando el documento de Excel en línea:

<https://riptutorial.com/es/epplus/topic/8202/guardando-el-documento-de-excel>

Capítulo 10: Importando datos desde archivo existente

Introducción

Cómo importar datos desde un archivo existente de Excel o CSV.

Examples

Importar datos desde archivo Excel

```
//create a list to hold all the values
List<string> excelData = new List<string>();

//read the Excel file as byte array
byte[] bin = File.ReadAllBytes("C:\\ExcelDemo.xlsx");

//or if you use asp.net, get the relative path
byte[] bin = File.ReadAllBytes(Server.MapPath("ExcelDemo.xlsx"));

//create a new Excel package in a memorystream
using (MemoryStream stream = new MemoryStream(bin))
using (ExcelPackage excelPackage = new ExcelPackage(stream))
{
    //loop all worksheets
    foreach (ExcelWorksheet worksheet in excelPackage.Workbook.Worksheets)
    {
        //loop all rows
        for (int i = worksheet.Dimension.Start.Row; i <= worksheet.Dimension.End.Row; i++)
        {
            //loop all columns in a row
            for (int j = worksheet.Dimension.Start.Column; j <=
worksheet.Dimension.End.Column; j++)
            {
                //add the cell data to the List
                if (worksheet.Cells[i, j].Value != null)
                {
                    excelData.Add(worksheet.Cells[i, j].Value.ToString());
                }
            }
        }
    }
}
```

Importar datos desde archivo CSV

```
//set the formatting options
ExcelTextFormat format = new ExcelTextFormat();
format.Delimiter = ';';
format.Culture = new CultureInfo(Thread.CurrentThread.CurrentCulture.ToString());
format.Culture.DateTimeFormat.ShortDatePattern = "dd-mm-yyyy";
```

```

format.Encoding = new UTF8Encoding();

//read the CSV file from disk
FileInfo file = new FileInfo("C:\\\\CSVDemo.csv");

//or if you use asp.net, get the relative path
FileInfo file = new FileInfo(Server.MapPath("CSVDemo.csv"));

//create a new Excel package
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //load the CSV data into cell A1
    worksheet.Cells["A1"].LoadFromText(file, format);
}

```

Importar datos desde un archivo de Excel con FileUpload Control

```

//check if there is actually a file being uploaded
if (FileUpload1.HasFile)
{
    //load the uploaded file into the memorystream
    using (MemoryStream stream = new MemoryStream(FileUpload1.FileBytes))
    using (ExcelPackage excelPackage = new ExcelPackage(stream))
    {
        //loop all worksheets
        foreach (ExcelWorksheet worksheet in excelPackage.Workbook.Worksheets)
        {
            //loop all rows
            for (int i = worksheet.Dimension.Start.Row; i <= worksheet.Dimension.End.Row; i++)
            {
                //loop all columns in a row
                for (int j = worksheet.Dimension.Start.Column; j <=
worksheet.Dimension.End.Column; j++)
                {
                    //add the cell data to the List
                    if (worksheet.Cells[i, j].Value != null)
                    {
                        excelData.Add(worksheet.Cells[i, j].Value.ToString());
                    }
                }
            }
        }
    }
}

```

Crear un DataTable desde un archivo de Excel

```

public static DataTable ExcelPackageToDataTable(ExcelPackage excelPackage)
{
    DataTable dt = new DataTable();
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets[1];

    //check if the worksheet is completely empty
    if (worksheet.Dimension == null)

```

```

{
    return dt;
}

//create a list to hold the column names
List<string> columnNames = new List<string>();

//needed to keep track of empty column headers
int currentColumn = 1;

//loop all columns in the sheet and add them to the datatable
foreach (var cell in worksheet.Cells[1, 1, 1, worksheet.Dimension.End.Column])
{
    string columnName = cell.Text.Trim();

    //check if the previous header was empty and add it if it was
    if (cell.Start.Column != currentColumn)
    {
        columnNames.Add("Header_" + currentColumn);
        dt.Columns.Add("Header_" + currentColumn);
        currentColumn++;
    }

    //add the column name to the list to count the duplicates
    columnNames.Add(columnName);

    //count the duplicate column names and make them unique to avoid the exception
    //A column named 'Name' already belongs to this DataTable
    int occurrences = columnNames.Count(x => x.Equals(columnName));
    if (occurrences > 1)
    {
        columnName = columnName + "_" + occurrences;
    }

    //add the column to the datatable
    dt.Columns.Add(columnName);

    currentColumn++;
}

//start adding the contents of the excel file to the datatable
for (int i = 2; i <= worksheet.Dimension.End.Row; i++)
{
    var row = worksheet.Cells[i, 1, i, worksheet.Dimension.End.Column];
    DataRow newRow = dt.NewRow();

    //loop all cells in the row
    foreach (var cell in row)
    {
        newRow[cell.Start.Column - 1] = cell.Text;
    }

    dt.Rows.Add(newRow);
}

return dt;
}

```

Lea **Importando datos desde archivo existente en línea:**

<https://riptutorial.com/es/epplus/topic/8290/importando-datos-desde-archivo-existente>

Capítulo 11: Mesas

Introducción

Este tema describe cómo agregar y diseñar tablas.

Examples

Añadiendo y formando una tabla

```
//Using statement for ExcelTable and TableStyles
using OfficeOpenXml.Table;

//Defining the tables parameters
int firstRow =1;
int lastRow = worksheet.Dimension.End.Row;
int firstColumn = 1;
int lastColumn = worksheet.Dimension.End.Column;
ExcelRange rg = worksheet.Cells[firstRow, firstColumn, lastRow, LastColumn];
string tableName = "Table1";

//Ading a table to a Range
ExcelTable tab = worksheet.Tables.Add(rg, tableName);

//Formating the table style
tab.TableStyle = TableStyles.Light8;
```

Lea Mesas en línea: <https://riptutorial.com/es/epplus/topic/8720/mesas>

Capítulo 12: Rellenando el documento con datos.

Introducción

Cómo llenar su hoja de Excel creada con datos de diferentes fuentes.

Examples

Rellenar con una tabla de datos

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a datatable
    DataTable dataTable = new DataTable();

    //add three columns to the datatable
    dataTable.Columns.Add("ID", typeof(int));
    dataTable.Columns.Add("Type", typeof(string));
    dataTable.Columns.Add("Name", typeof(string));

    //add some rows
    dataTable.Rows.Add(0, "Country", "Netherlands");
    dataTable.Rows.Add(1, "Country", "Japan");
    dataTable.Rows.Add(2, "Country", "America");
    dataTable.Rows.Add(3, "State", "Gelderland");
    dataTable.Rows.Add(4, "State", "Texas");
    dataTable.Rows.Add(5, "State", "Echizen");
    dataTable.Rows.Add(6, "City", "Amsterdam");
    dataTable.Rows.Add(7, "City", "Tokyo");
    dataTable.Rows.Add(8, "City", "New York");

    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add all the content from the DataTable, starting at cell A1
    worksheet.Cells["A1"].LoadFromDataTable(dataTable, true);
}
```

Rellene con una tabla de datos desde una consulta SQL o un procedimiento almacenado

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //the query or stored procedure name for the database
    string sqlQuery = "SELECT * FROM myTable";

    //create a datatable
    DataTable dataTable = loadExternalDataSet(sqlQuery);
}
```

```

//create a WorkSheet
ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

//add all the content from the DataTable, starting at cell A1
worksheet.Cells["A1"].LoadFromDataTable(dataTable, true);
}

//method for retrieving data from the database and return it as a datatable
public static DataTable loadExternalDataSet(string sqlQuery)
{
    DataTable dt = new DataTable();

    using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["myConnStr"].ConnectionString))
    using (SqlDataAdapter adapter = new SqlDataAdapter(sqlQuery, connection))
    {
        try
        {
            adapter.Fill(dt);
        }
        catch
        {
        }
    }

    return dt;
}

```

Rellenar manualmente las celdas

Rellena algunas celdas con texto.

```

worksheet.Cells["A1"].Value = "Lorem ipsum";
worksheet.Cells["B2"].Value = "dolor sit amet";
worksheet.Cells["C3"].Value = "consectetur adipiscing";
worksheet.Cells["D4"].Value = "elit sed do eiusmod";

worksheet.Cells["E5"].Value = 12345;
worksheet.Cells["F6"].Value = DateTime.Now;

```

Rellene datos de celda con un bucle, tenga en cuenta que los índices de fila y columna comienzan en 1

```

for (int i = 1; i <= 30; i++)
{
    for (int j = 1; j <= 15; j++)
    {
        worksheet.Cells[i, j].Value = "Row " + i + ", Column " + j;
    }
}

```

Rellenar desde colección

```

//create a new ExcelPackage

```

```
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //create a new list with books
    List<Book> books = new List<Book>();

    //add some books to the list
    for (int i = 0; i < 10; i++)
    {
        Book b = new Book();

        b.id = i;
        b.name = "Name " + i;
        b.category = "Category " + i;
        b.date = DateTime.Now.AddDays(i).AddHours(i);

        books.Add(b);
    }

    //add all the content from the List<Book> collection, starting at cell A1
    worksheet.Cells["A1"].LoadFromCollection(books);
}
```

Lea [Rellenando el documento con datos. en línea:](https://riptutorial.com/es/epplus/topic/8223/rellenando-el-documento-con-datos-)

<https://riptutorial.com/es/epplus/topic/8223/rellenando-el-documento-con-datos->

Capítulo 13: Tabla dinámica

Introducción

La tabla dinámica es un tipo de tabla interactiva, que se puede utilizar para calcular datos, como obtener datos de suma o recuento. Además, los usuarios pueden cambiar el diseño de la tabla dinámica para analizar datos de diferentes maneras o reasignar la etiqueta de fila / columna. Cada vez que los usuarios cambian de diseño, los datos se volverán a calcular en la tabla dinámica.

Examples

Creación de una tabla dinámica

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create 2 WorkSheets. One for the source data and one for the Pivot table
    ExcelWorksheet worksheetPivot = excelPackage.Workbook.Worksheets.Add("Pivot");
    ExcelWorksheet worksheetData = excelPackage.Workbook.Worksheets.Add("Data");

    //add some source data
    worksheetData.Cells["A1"].Value = "Column A";
    worksheetData.Cells["A2"].Value = "Group A";
    worksheetData.Cells["A3"].Value = "Group B";
    worksheetData.Cells["A4"].Value = "Group C";
    worksheetData.Cells["A5"].Value = "Group A";
    worksheetData.Cells["A6"].Value = "Group B";
    worksheetData.Cells["A7"].Value = "Group C";
    worksheetData.Cells["A8"].Value = "Group A";
    worksheetData.Cells["A9"].Value = "Group B";
    worksheetData.Cells["A10"].Value = "Group C";
    worksheetData.Cells["A11"].Value = "Group D";

    worksheetData.Cells["B1"].Value = "Column B";
    worksheetData.Cells["B2"].Value = "emc";
    worksheetData.Cells["B3"].Value = "fma";
    worksheetData.Cells["B4"].Value = "h2o";
    worksheetData.Cells["B5"].Value = "emc";
    worksheetData.Cells["B6"].Value = "fma";
    worksheetData.Cells["B7"].Value = "h2o";
    worksheetData.Cells["B8"].Value = "emc";
    worksheetData.Cells["B9"].Value = "fma";
    worksheetData.Cells["B10"].Value = "h2o";
    worksheetData.Cells["B11"].Value = "emc";

    worksheetData.Cells["C1"].Value = "Column C";
    worksheetData.Cells["C2"].Value = 299;
    worksheetData.Cells["C3"].Value = 792;
    worksheetData.Cells["C4"].Value = 458;
    worksheetData.Cells["C5"].Value = 299;
    worksheetData.Cells["C6"].Value = 792;
    worksheetData.Cells["C7"].Value = 458;
}
```



```

worksheetData.Cells["C8"].Value = 299;
worksheetData.Cells["C9"].Value = 792;
worksheetData.Cells["C10"].Value = 458;
worksheetData.Cells["C11"].Value = 299;

worksheetData.Cells["D1"].Value = "Column D";
worksheetData.Cells["D2"].Value = 40075;
worksheetData.Cells["D3"].Value = 31415;
worksheetData.Cells["D4"].Value = 384400;
worksheetData.Cells["D5"].Value = 40075;
worksheetData.Cells["D6"].Value = 31415;
worksheetData.Cells["D7"].Value = 384400;
worksheetData.Cells["D8"].Value = 40075;
worksheetData.Cells["D9"].Value = 31415;
worksheetData.Cells["D10"].Value = 384400;
worksheetData.Cells["D11"].Value = 40075;

//define the data range on the source sheet
var dataRange = worksheetData.Cells[worksheetData.Dimension.Address];

//create the pivot table
var pivotTable = worksheetPivot.PivotTables.Add(worksheetPivot.Cells["B2"], dataRange,
"PivotTable");

//label field
pivotTable.RowFields.Add(pivotTable.Fields["Column A"]);
pivotTable.DataOnRows = false;

//data fields
var field = pivotTable.DataFields.Add(pivotTable.Fields["Column B"]);
field.Name = "Count of Column B";
field.Function = DataFieldFunctions.Count;

field = pivotTable.DataFields.Add(pivotTable.Fields["Column C"]);
field.Name = "Sum of Column C";
field.Function = DataFieldFunctions.Sum;
field.Format = "0.00";

field = pivotTable.DataFields.Add(pivotTable.Fields["Column D"]);
field.Name = "Sum of Column D";
field.Function = DataFieldFunctions.Sum;
field.Format = "€#,##0.00";
}

```

Lea Tabla dinámica en línea: <https://riptutorial.com/es/epplus/topic/8767/tabla-dinamica>

Capítulo 14: Texto enriquecido en celdas

Introducción

La mayoría de las veces, cuando creamos hojas de cálculo, solo usamos la propiedad Valor de la celda para poner contenido en la celda y la propiedad Estilo para formatearla.

De vez en cuando, sin embargo, es posible que desee aplicar varios estilos a una celda, tal vez colocar un título en negrita y subrayado antes del resto del contenido, o resaltar una parte particular del texto en rojo: aquí es donde entra en juego la propiedad RichText de la celda. .

Examples

Añadiendo RichText a una celda

Cada elemento del texto en el que desea utilizar un formato distinto debe agregarse por separado, agregándolo a la propiedad de la colección RichText de la celda.

```
var cell = ws.Cells[1,1];
cell.IsRichText = true; // Cell contains RichText rather than basic values
cell.Style.WrapText = true; // Required to honor new lines

var title = cell.RichText.Add("This is my title");
var text = cell.RichText.Add("\nAnd this is my text");
```

Tenga en cuenta que cada vez que agregue () una nueva cadena, heredará el formato de la sección anterior. Como tal, si desea cambiar el formato predeterminado, solo tendrá que cambiarlo en la primera cadena agregada.

Sin embargo, este comportamiento puede causar cierta confusión al formatear su texto. Usando el ejemplo anterior, el siguiente código hará que **todo el texto** de la celda esté en negrita y cursiva; este no es el comportamiento deseado:

```
// Common Mistake
var title = cell.RichText.Add("This is my title");
title.Bold = true;
title.Italic = true;

var text = cell.RichText.Add("\nAnd this is my text"); // Will be Bold and Italic too
```

El enfoque preferido es agregar todas las secciones de texto primero, luego aplicar el formato específico de la sección después, como se muestra aquí:

```
var title = cell.RichText.Add("This is my title");
title.FontName = "Verdana"; // This will be applied to all subsequent sections as well

var text = cell.RichText.Add("\nAnd this is my text");
```

```
// Format JUST the title
title.Bold = true;
title.Italic = true;
```

Propiedades de formato de texto

Hay una serie de propiedades que se pueden aplicar a las secciones de RichText.

```
var title = cell.RichText.Add("This is my title");

// Data Type:      bool
// Default Value:  false
title.Bold = true;

// Data Type:      System.Drawing.Color
// Default Value:  Color.Black
title.Color = Color.Red;
title.Color = Color.FromArgb(255, 0, 0);
title.Color = ColorTranslator.FromHtml("#FF0000");

// Data Type:      string
// Default Value:  "Calibri"
title.FontName = "Verdana";

// Data Type:      bool
// Default Value:  false
title.Italic = true;

// Data Type:      bool
// Default Value:  true
// If this property is set to false, any whitespace (including new lines)
// is trimmed from the start and end of the Text
title.PreserveSpace = true;

// Data Type:      float
// Default Value:  11
// The font size is specified in Points
title.Size = 16;

// Data Type:      bool
// Default Value:  false
// Strikethrough
title.Strike = false;

// Data Type:      string
// Default Value:  Whatever was set when the text was added to the RichText collection
title.Text += " (updated)";

// Data Type:      bool
// Default Value:  false
title.UnderLine = true;

// Data Type:      OfficeOpenXml.Style.ExcelVerticalAlignmentFont
// Default Value:  ExcelVerticalAlignmentFont.None
title.VerticalAlign = ExcelVerticalAlignmentFont.None;
```

Insertando RichText en una celda

EPPlus también admite la capacidad de insertar texto en una celda mediante el método `Insert()`. Por ejemplo:

```
var file = new FileInfo(filePath);
using (var p = new ExcelPackage(file))
{
    var wb = p.Workbook;
    var ws = wb.Worksheets.FirstOrDefault() ?? wb.Worksheets.Add("Sheet1");

    var cell = ws.Cells[1, 1];
    cell.IsRichText = true;
    cell.RichText.Clear(); // Remove any RichText that may be in the cell already
    var s1 = cell.RichText.Add("Section 1.");
    var s2 = cell.RichText.Add("Section 2.");

    var s3 = cell.RichText.Insert(1, "Section 3.");

    s3.Bold = true;
    p.Save();
}
```

Tenga en cuenta que el método `Insert()` NO se inserta en un índice de caracteres, sino en un índice de Sección. Debido a que las secciones están indexadas a cero, el código anterior producirá el siguiente texto en la celda:

Sección 1. Sección 3. Sección 2.

Lea [Texto enriquecido en celdas en línea](https://riptutorial.com/es/epplus/topic/10776/texto-enriquecido-en-celdas): <https://riptutorial.com/es/epplus/topic/10776/texto-enriquecido-en-celdas>

Capítulo 15: Validación de entrada de usuario

Introducción

Cómo validar las entradas de los usuarios. La validación restringe los valores que un usuario puede ingresar en una celda y / o establece un cuadro combinado para que el usuario seleccione el valor de la celda. Opcionalmente, se puede mostrar un mensaje cuando el usuario hace clic en una celda y un mensaje de error, cuando falla la validación.

Examples

Validación de lista

```
//Add a List validation to B column. Values should be in a list
var val = worksheet.DataValidations.AddListValidation("B:B");
//Shows error message when the input doesn't match the accepted values
val.ShowErrorMessage = true;
//Style of warning. "information" and "warning" allow users to ignore the validation,
//while "stop" and "undefined" doesn't
val.ErrorStyle = OfficeOpenXml.DataValidation.ExcelDataValidationWarningStyle.information;
//Title of the error message box
val.ErrorTitle = "This is the title";
//Message of the error
val.Error = "This is the message";
//Set to true to show a prompt when user clics on the cell
val.ShowInputMessage = true;
//Set the message for the prompt
val.Prompt = "This is a input message";
//Set the title for the prompt
val.PromptTitle = "This is the title from the input message";
//Define the accepted values
val.Formula.Values.Add("This is accepted");
val.Formula.Values.Add("This is also accepted");
val.Formula.Values.Add("Any other thing is rejected");
//Set to true if blank value is accepted
val.AllowBlank = false;

//Add a List validation to the C column
var val2 = worksheet.DataValidations.AddListValidation("C:C");
//Define the Cells with the accepted values
val2.Formula.ExcelFormula = "=$D$3:$D$5";
//Fill the cells with the accepted values
worksheet.Cells["D3"].Value = "Val1";
worksheet.Cells["D4"].Value = "Val2";
worksheet.Cells["D5"].Value = "Val3";
```

Validación de enteros

```
//Add a List validation to the C column
var val3 = worksheet.DataValidations.AddIntegerValidation("E:E");
//For Integer Validation, you have to set error message to true
val3.ShowErrorMessage = true;
```

```
val3.Error = "The value must be an integer between 0 and 10";  
//Minimum allowed Value  
val3.Formula.Value = 0;  
//Maximum allowed Value  
val3.Formula2.Value = 10;  
//If the cells are not filled, allow blanks or fill with a valid value,  
//otherwise it could generate a error when saving  
val3.AllowBlank = true;
```

Validación de fecha y hora

```
//Add a DateTime Validation to column F  
var val4 = worksheet.DataValidations.AddDateTimeValidation("F:F");  
//For DateTime Validation, you have to set error message to true  
val4.ShowErrorMessage = true;  
//Minimum allowed date  
val4.Formula.Value = new DateTime(2017,03,15, 01, 0,0);  
//Maximum allowed date  
val4.Formula2.Value= new DateTime(2017, 03, 16, 12, 0, 0);  
val4.AllowBlank = true;
```

Validación de la longitud del texto

```
//Add a TextLength Validation to column G  
var val5 = worksheet.DataValidations.AddTextLengthValidation("G:G");  
//For TextLenght Validation, you have to set error message to true  
val5.ShowErrorMessage = true;  
//Minimum allowed text lenght  
val5.Formula.Value = 3;  
//Maximum allowed text lenght  
val5.Formula2.Value = 5;  
val5.AllowBlank = true;
```

Lea Validación de entrada de usuario en línea:

<https://riptutorial.com/es/epplus/topic/8739/validacion-de-entrada-de-usuario>

Creditos

S. No	Capítulos	Contributors
1	Empezando con epplus	Community , Magnetron , VDWWD
2	Adjuntar datos al documento existente	VDWWD
3	Columnas y filas	hellyale , Magnetron , VDWWD
4	Combinar células	Magnetron
5	Creando cuadros	VDWWD
6	Creando fórmulas y calculando rangos.	Magnetron , VDWWD
7	Diseño del documento de Excel	Magnetron , VDWWD
8	Formato de valores	Magnetron , VDWWD
9	Guardando el documento de Excel	Magnetron , VDWWD
10	Importando datos desde archivo existente	VDWWD
11	Mesas	Magnetron
12	Rellenando el documento con datos.	VDWWD
13	Tabla dinámica	VDWWD
14	Texto enriquecido en celdas	Pete
15	Validación de entrada de usuario	Magnetron