



Benemérita Universidad Autónoma de Puebla

Materia: *Sistemas Operativos II*

Profesor: *Gilberto López Poblano*

Alumno: *Christian Amauri Amador Ortega - 201927821*

- Word rtf -

Para esta serie de prácticas con código utilizamos el lenguaje C# en el IDE visual studio (la versión más reciente en septiembre de 2022). Puede haber variaciones en el funcionamiento de exactamente los mismos códigos pero con otra versión de visual studio u otro IDE. De todas formas estas variaciones deben de ser mínimas e insignificantes.

*!!! Documentaremos el funcionamiento de las aplicaciones (en su ejecución) y haremos algunos comentarios / notas / consideraciones / advertencias al respecto. Para ahorrar tiempo y esfuerzo, trataremos de no ahondar demasiado en la estructura del código, sino simplemente explicar brevemente cómo funciona **(se asume que el lector tiene ya conocimientos suficientes sobre conceptos varios de programación estructurada, ambiente gráfico, variables, datos, y demás...)***

Finalmente hay que tomar en cuenta que C# y Visual studio son herramientas pesadas y caprichosas. Y que el equipo que vayamos a utilizar debe tener cierta capacidad de rendimiento superior a sólo básica.

Primero, el objetivo del proyecto es crear un programa en C# con el que podamos convertir documentos tipo Word (.docx) a documentos tipo .rft

En cualquier caso, el repositorio en el que este código se encuentra (y otros códigos en C# del curso de Sistemas operativos 2, BUAP FCC, Otoño 2022) está disponible haciendo click en el siguiente enlace:

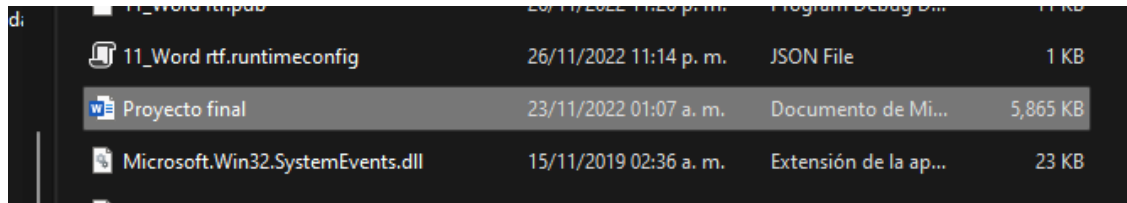
<https://drive.google.com/drive/folders/1yZujMI51XAnEZBMj-4ykUWMfHXBC9uYp?usp=sharing>

Y estará disponible al menos durante el resto de 2022. Gracias. 😊

Ejemplo rápido del funcionamiento general del proyecto, para resumir:

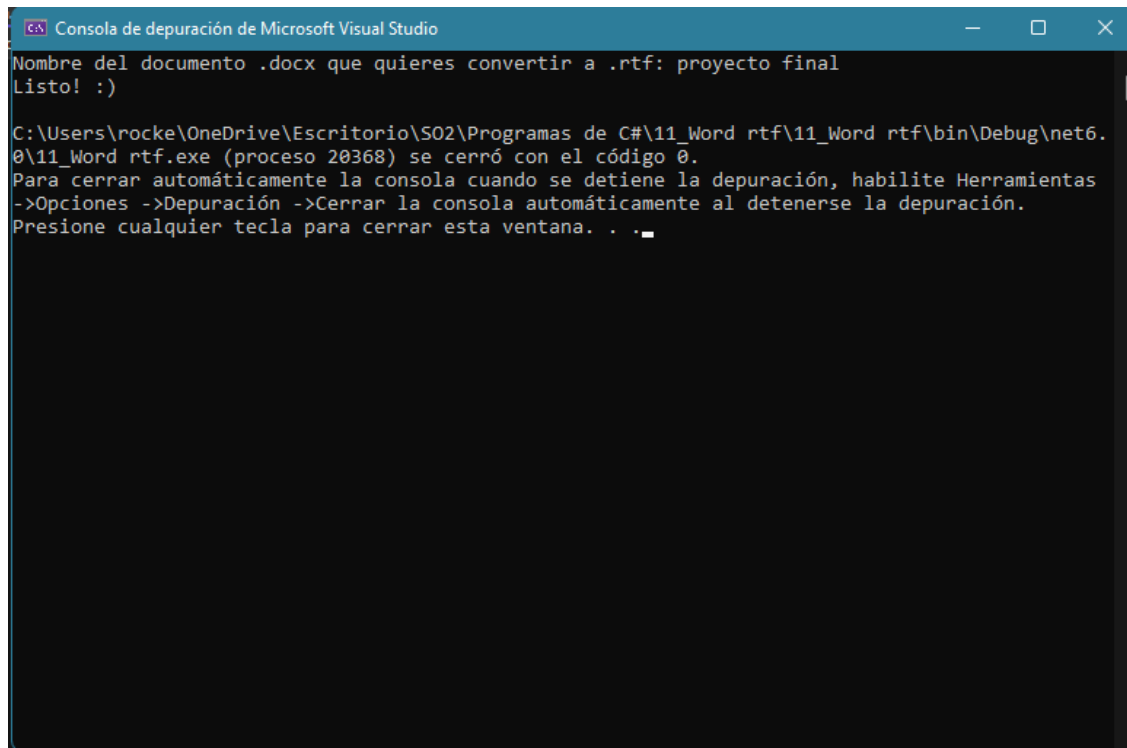
- Podemos crear una copia de un documento con extensión .docx pero con extensión .rtf con una sola línea en una aplicación de consola:

Antes:



11_Word rtf.runtimeconfig	26/11/2022 11:14 p. m.	JSON File	1 KB
Proyecto final	23/11/2022 01:07 a. m.	Documento de Mi...	5,865 KB
Microsoft.Win32.SystemEvents.dll	15/11/2019 02:36 a. m.	Extensión de la ap...	23 KB

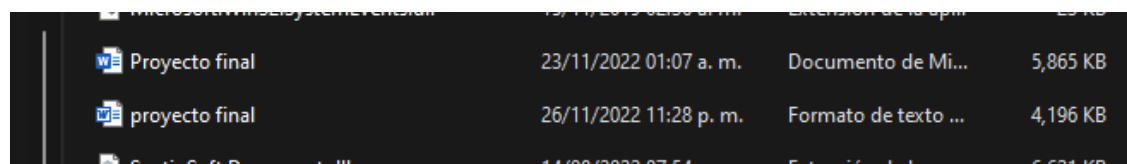
Ejecución:



```
Consola de depuración de Microsoft Visual Studio
Nombre del documento .docx que quieres convertir a .rtf: proyecto final
Listo! :)

C:\Users\rocke\OneDrive\Escritorio\S02\Programas de C#\11_Word rtf\11_Word rtf\bin\Debug\net6.0\11_Word rtf.exe (proceso 20368) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

Después:



Proyecto final	23/11/2022 01:07 a. m.	Documento de Mi...	5,865 KB
proyecto final	26/11/2022 11:28 p. m.	Formato de texto ...	4,196 KB
Microsoft.Win32.SystemEvents.dll	15/11/2019 02:36 a. m.	Extensión de la ap...	23 KB

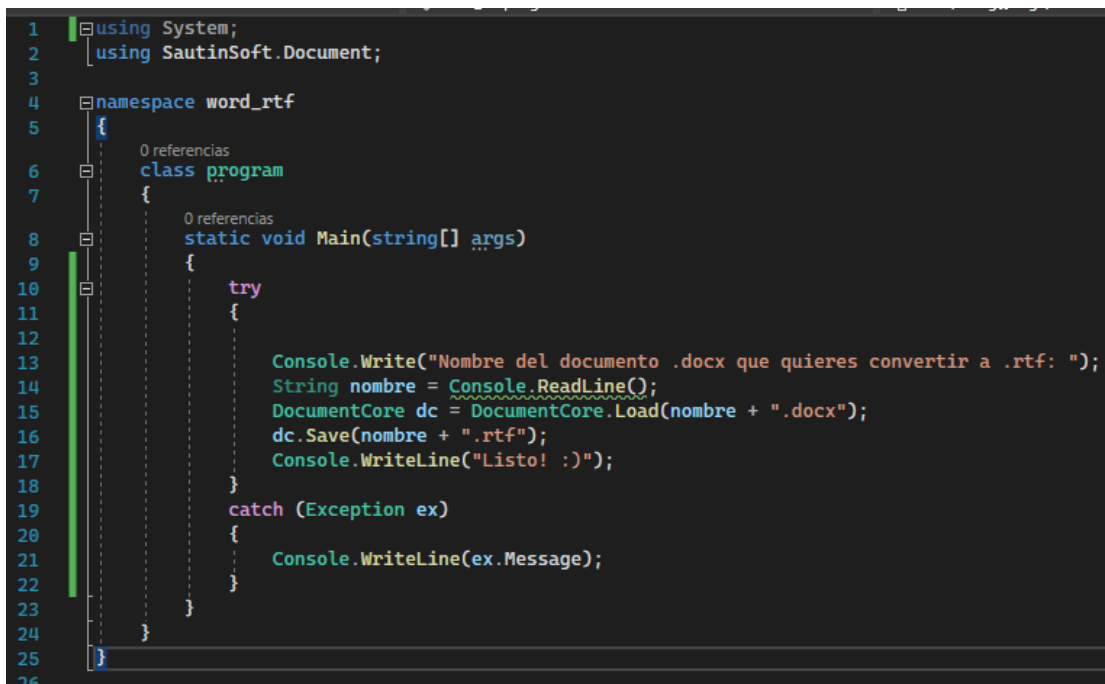
Ahora, los archivos .cs del proyecto (que nosotros creamos) corresponde únicamente a un script:

- Script -

Program.cs

La estructura inicial de este script consta de:

- 2 librerías
- namespace definido como: Word_rtf
- class program > static void main()



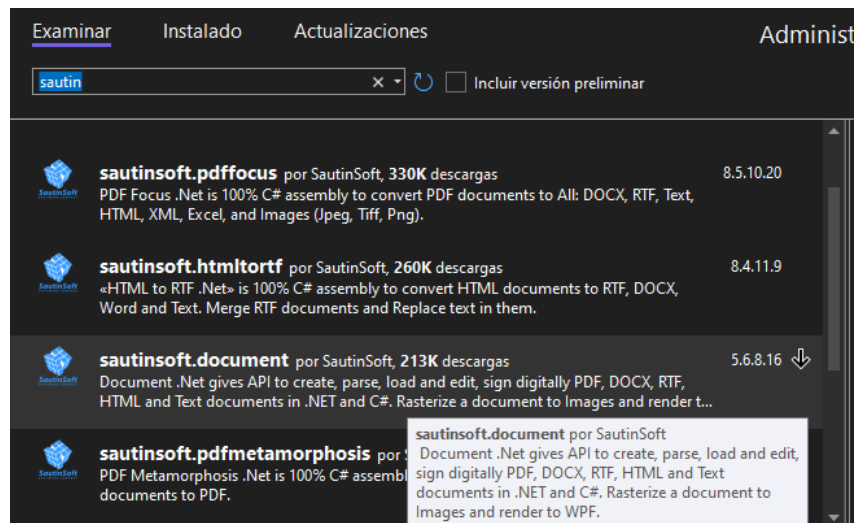
```
1 using System;
2 using SautinSoft.Document;
3
4 namespace word_rtf
5 {
6     class program
7     {
8         static void Main(string[] args)
9         {
10             try
11             {
12                 Console.Write("Nombre del documento .docx que quieres convertir a .rtf: ");
13                 String nombre = Console.ReadLine();
14                 DocumentCore dc = DocumentCore.Load(nombre + ".docx");
15                 dc.Save(nombre + ".rtf");
16                 Console.WriteLine("Listo! :)");
17             }
18             catch (Exception ex)
19             {
20                 Console.WriteLine(ex.Message);
21             }
22         }
23     }
24 }
25
26
```

Como podemos apreciar, la estructura del programa es exageradamente simple, simplemente obtenemos por consola el nombre del documento .docx del que queremos crear una copia en .rtf mediante una variable tipo **String**, y luego usamos ese nombre para llamar a las funciones **Load()** y **Save()** de la librería **SautinSoft.Document**.

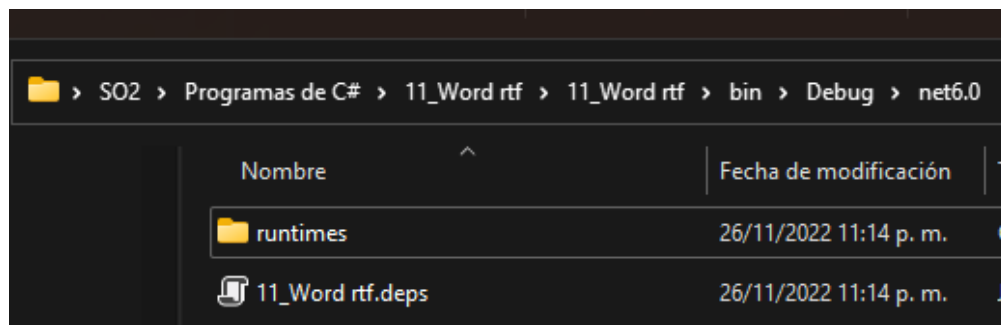
Notemos que **DocumentCore** es una clase “documento”, que debe recibir como parámetros el nombre del documento, junto con su formato (en este caso, .docx).

Y que para crear la copia que queremos, hay que usar su método **Save()** junto con el nombre que le queremos dar a esa copia (en este caso el mismo) y su extensión (en este caso, .rtf)

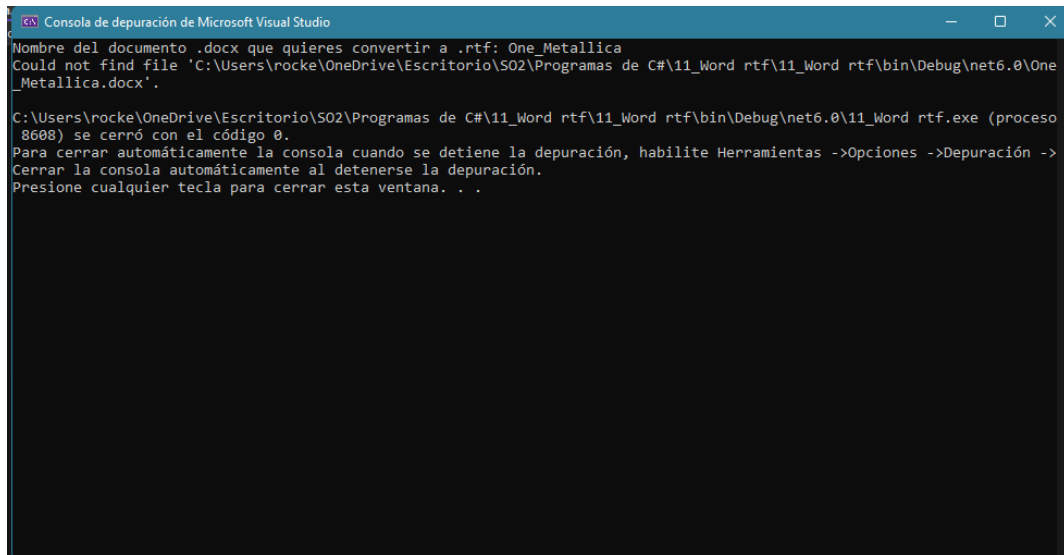
Hay que aclarar que esta librería (**SautinSoft.Document**) debe ser descargada aparte, en el apartado de administrar paqueterías NuGet:



Para que el documento pueda ser encontrado por el programa, este debe estar en la carpeta bin > debug > net6.0 de la solución de nuestro programa C#:



En caso de que en esa carpeta no se encuentre el documento deseado, simplemente se mostrará en la consola el mensaje de la excepción mencionada:



```
Consola de depuración de Microsoft Visual Studio
Nombre del documento .docx que quieres convertir a .rtf: One_Metallica
Could not find file 'C:\Users\rocke\OneDrive\Escritorio\S02\Programas de C#\11_Word rtf\11_Word rtf\bin\Debug\net6.0\One_Metallica.docx'.

C:\Users\rocke\OneDrive\Escritorio\S02\Programas de C#\11_Word rtf\11_Word rtf\bin\Debug\net6.0\11_Word rtf.exe (proceso 8608) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración -> Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

Con estas bases, podríamos crear una aplicación más vistosa, con ambiente gráfico y quizá más opciones, pero por cuestiones de tiempo y didácticas, deberá bastarnos con esto. De todas formas el objetivo está más que cumplido y es bastante sencillo.