ISIL APRENDE HACIENDO

TITULO

DESARROLLO DE APLICACIONES I

Sesión Nro. 7 (Parte 2)

Objetivos de la sesión:

Al culminar la presente podrás:

- **Aprender a generar de archivos de reportes:**
  - **Archivos Excel**
  - **Archivos XML**

- **Conocer como se puede realizar una tarea asíncrona con el control BackgrounWorker**

Temario

1. Creando archivos Excel con la librería EPPlus
2. Manejando el control BackgroundWorker
3. Interactuando con archivos XML : métodos ReadXml y WriteXml

Tema Nro. 1:
Creando listados con archivos EXCEL y la Librería EPPLUS

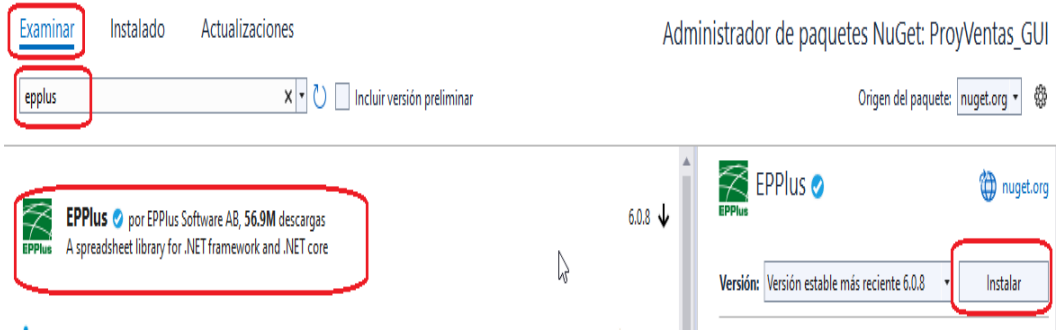
1.1 Creando archivos Excel con EPPlus

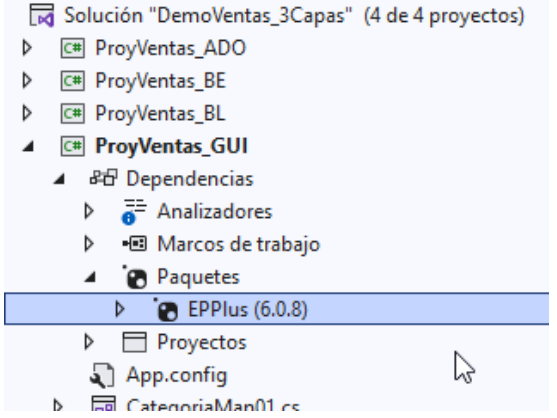
EPPlus es una biblioteca .NET que lee y escribe archivos de Excel 2007 o posterior utilizando el formato Open Office Xml (xlsx). EPPlus soporta:

- **Rangos, Estilo de celda (borde, color, relleno, fuente, número, alineaciones)**
- **Imágenes, Comentarios ,Protección, Cifrado, Tablas dinámicas, Validación de datos , Formato condicional**
- **VBA**

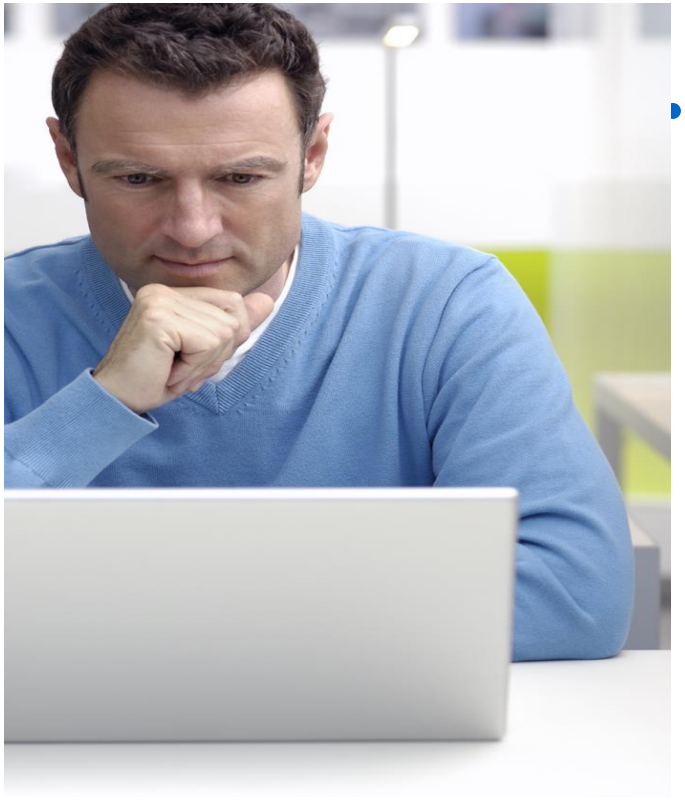
1.1 Creando archivos Excel con EPPlus

Para utilizar EPPlus, lo debe agregar desde el Administrador de Paquetes Nuget como ya se hizo en otros casos, buscando el paquete EPPlus e instalándolo, da tal manera que el proyecto "ProyVentas_GUI" tenga esta vista:





Demostración : Creando archivos Excel con EPPLUS



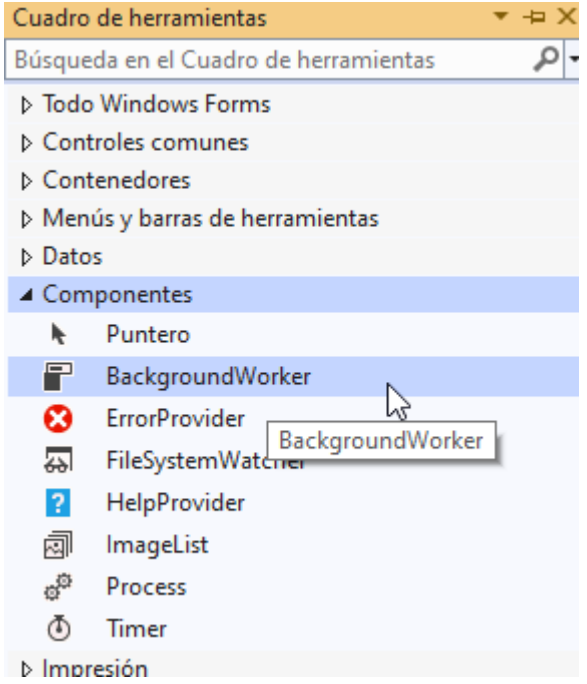
- En esta demostración, aprenderemos a generar listados Excel empleando la librería EPPLUS.

Tema Nro. 2:
El control BackgroundWorker

2.1 El control BackgroundWorker

Este control permite hacer trabajos asíncronos, de tal manera que mientras se ejecute una tarea podamos hacer otra, sin necesidad de esperar que termine la primera tarea encomendada.

Si le asignamos alguna tarea a este control , podemos incluso mostrar al usuario el grado de avance de la tarea (control ProgressBar) y poder mostrar un mensaje o icono de ejecución, para que si la tarea demora mucho, el usuario no piense que el sistema “no esta haciendo nada”, sino que por medio de un archivo “.gif” podamos indicar que se esta ejecutando un trabajo



2.1 El control BackgroundWorker

Es importante establecer que control delegara el trabajo al BackgroundWorker. Normalmente lo hacemos en el evento clic de un botón.

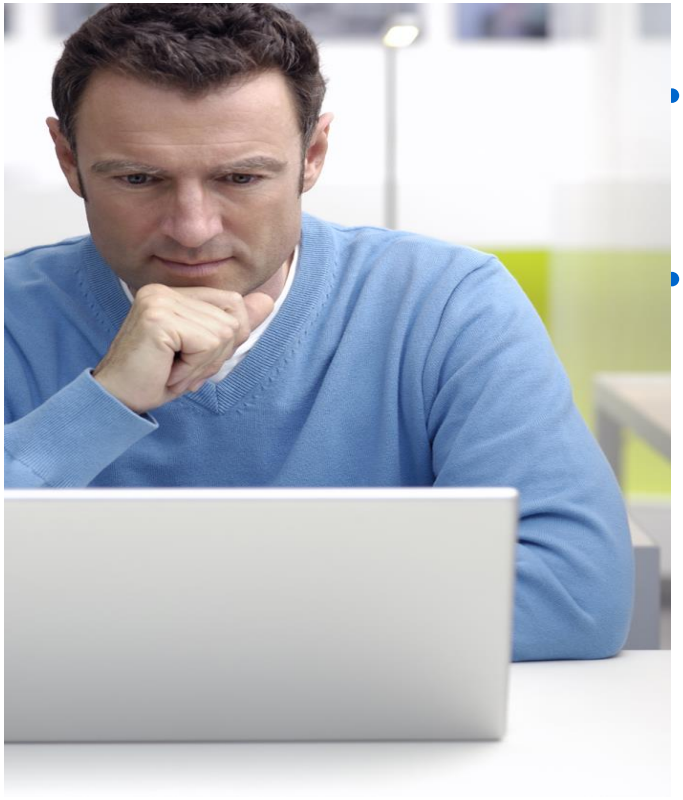
Este control maneja 3 eventos importantes:

a) DoWork: En donde debemos codificar el código que esta asociado al trabajo que debe hacer el control

b) ProgressChanged: En donde se va indicando el avance del trabajo encomendado al control. Para ello debe definir en true la propiedad “WorkerReportProgress”, de lo contrario el evento no se manejara.

c) RunWorkerCompleted: Evento que se dispara cuando el trabajo encomendado al control concluye.

Demostración : Empleando el control BackgroundWorker



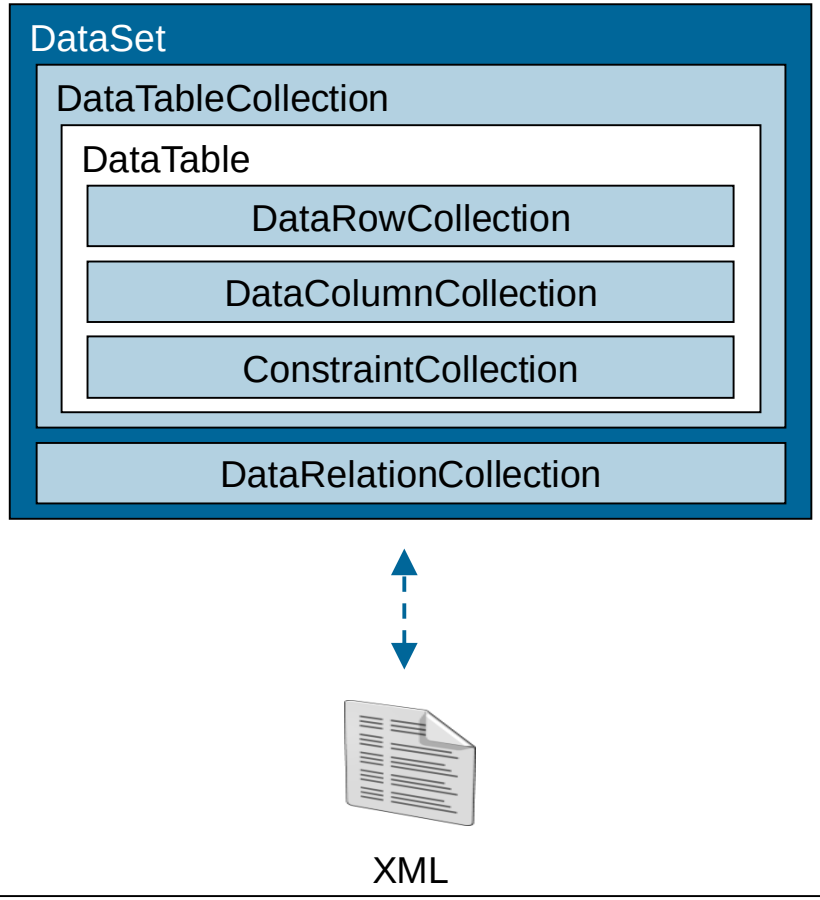
- En esta demostración, aprenderemos a emplear el control BackgroundWorker para una descarga de un archivo Excel.
- Desarrolle el ejercicio respectivo (Parte 1) detallado en la Guía Practica de la Semana 7. Ponga la atención debida a las especificaciones.

Tema Nro. 3:
Interacción con documentos XML y

3.1 Interaccion con documentos XML

- La información almacenada en un objeto Dataset como ya se explico esta en memoria del cliente, pero se puede guardar o recuperar en un archivo con formato XML gracias a 2 métodos que poseen los Dataset:
  - ReadXML : Que permite leer un documento XML y volcar toda la data dentro de un objeto Dataset
  - WriteXML: Que permite guardar la data de un objeto Dataset en un archivo XML

3.1 Interacción con documentos XML



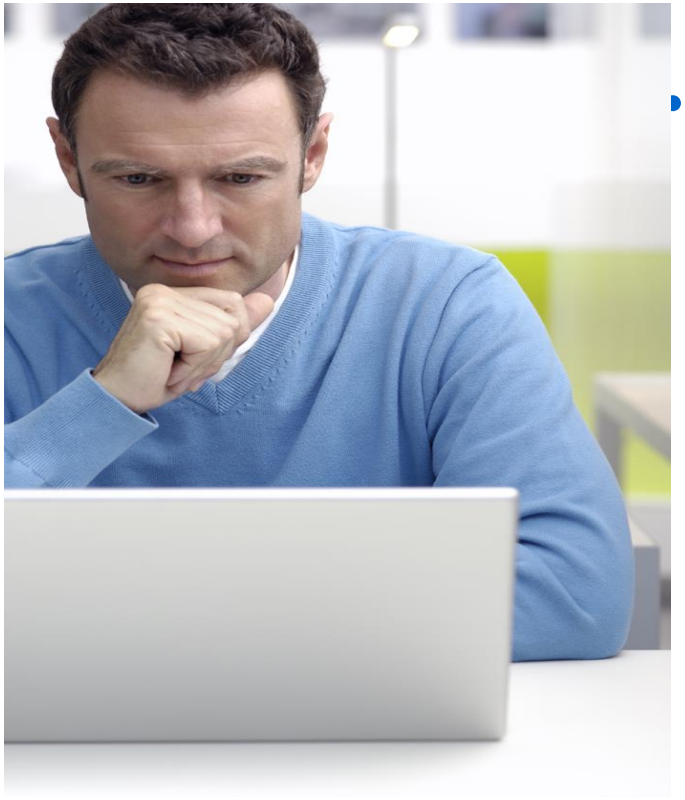
3.2 Escribiendo documentos y esquemas XML

```
try
{
   DataSet dts = new DataSet();
   dts = mifac.ListarFacturaDetalle();
   //Creación del archivo XML
   dts.WriteXml(@"D:\XML\Facturacion.xml");
   dts.WriteXmlSchema(@"D:\XML\EsquemaFact.xsd");
   // Ahora escribimos el XML en un StringWriter  y lo mostramos en la caja de texto
   System.IO.StringWriter swXML = new System.IO.StringWriter();
   dts.WriteXml(swXML);
   txtXML.Text = swXML.ToString();
   // Ahora escribimos el esquema XML en un StringWriter y lo mostramos en la caja de texto
   System.IO.StringWriter esXml = new System.IO.StringWriter();
   dts.WriteXmlSchema(esXml);
   txtSCH.Text = esXml.ToString();
}
catch (Exception ex)
{
   MessageBox.Show(ex.Message);
 }
```

3.3 Leyendo documentos y esquemas XML

```
try
 {
  DataSet dts = new DataSet();
  dts.ReadXmlSchema(@"D:\XML\EsquemaFact.xsd");
  dts.ReadXml(@"D:\XML\Facturacion.xml");
   dataGrid1.DataSource = dts.Tables["MisClientes"];
 }
 catch ( Exception ex)
 {
    MessageBox.Show(ex.Message);
 }
```

Demostración : Generando listados con archivos XML



- En esta demostración, conoceremos como crear listados con archivos XML y Visual Studio. Para ello desarrolle el ejercicio respectivo (Parte 2) de la Guía Practica de la semana 7. Ponga la atención debida a las especificaciones.

Conclusiones de la sesión:

- También aprendimos a crear listados en formatos de libro Excel gracias al empleo de la librería EPPlus.
- Por ultimo pudimos implementar tareas asíncronas mediante el control BackgroundWorker y medir el progreso de un trabajo
- Hemos aprendido a construir documentos XML que permiten la compartición de datos entre aplicaciones sin necesidad de estar conectados a una base de datos.