

Capítulo 16

Tecnologías complementarias en .NET

Objetivo

Al finalizar el capítulo, el alumno:

- Publicar reportes.
- Integrar reportes en formularios Windows.
- Reconocer los tipos de aplicaciones que pueden integrarse a Office.
- Crear un componente MS-Excel

Temas

1. Diseño de reportes
2. Crystal Reports
3. Reporting Services
4. Desarrollo de soluciones Office
5. Tipos de soluciones Office

1. Diseño de reportes

Diseño de reportes

Visual Studio, tiene dos opciones para generar reportes:

- Crystal Reports: es una aplicación de inteligencia empresarial utilizada para diseñar y generar reportes desde una amplia gama de fuentes de datos.
- Reporting Services: es una plataforma de reportes basada en servidor que proporciona la funcionalidad completa de generación de reportes para una gran variedad de orígenes de datos.

Para el diseño y posterior elaboración de reportes se tienen varias alternativas (Telerik Reporting, DevExpress Reporting, Crystal Reports, Reporting Services, etc.), entre todas ellas se van a tratar las 2 alternativas más populares:

- Crystal Reports es una aplicación de inteligencia empresarial utilizada para diseñar y generar reportes desde una amplia gama de fuentes de datos.
- Reporting Services es una plataforma de reportes basada en servidor que proporciona la funcionalidad completa de generación de reportes para una gran variedad de orígenes de datos.

2. Crystal Reports

Crystal Reports

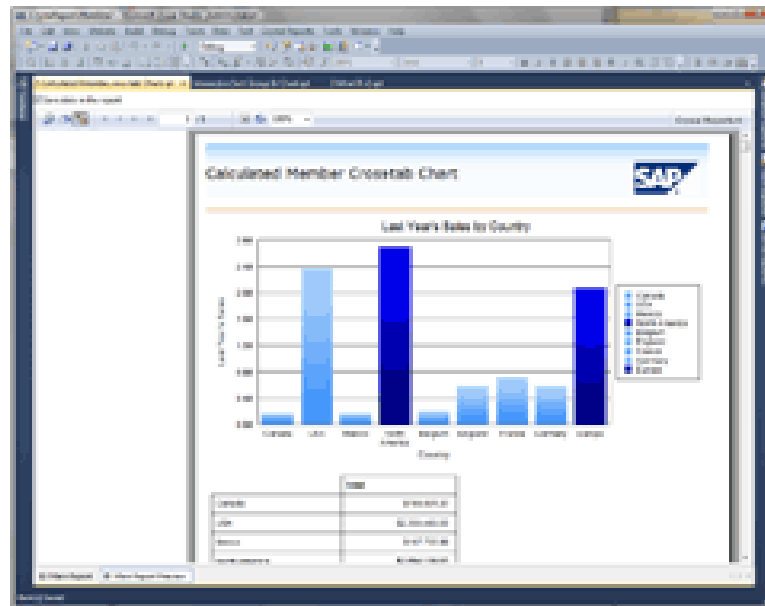
- Crystal Reports: es una aplicación de inteligencia empresarial utilizada para diseñar y generar reportes desde una amplia gama de fuentes de datos (bases de datos).



Crystal Reports, es una aplicación de inteligencia empresarial utilizada para diseñar y generar reportes desde una amplia gama de fuentes de datos (bases de datos).

El producto fue originalmente creado por Crystal Services Inc., la cual produjo las versiones 1.0 hasta la 3.0. La compañía fue subsecuentemente adquirida por Seagate Software, la que luego fue renombrada como Crystal Decisions, y produjo las versiones 4.0 hasta la 9.0. Crystal Decisions fue adquirida en diciembre de 2003 por Business Objects, la cual ha producido las versiones 10 y 11 (XI), y actualmente la versión 12 (2008). Business Objects fue adquirida el 8 de Octubre de 2007 por SAP.





Actualmente, Crystal Reports no viene en los programas de instalación de Visual Studio, es necesario descargarlo y agregarlo.

Objetos de reportes

Crystal Reports utiliza una funcionalidad de arrastrar y colocar, parecida a la que se utiliza en Visual Studio .NET. Se arrastra un objeto de reporte hasta el diseñador (como un campo de base de datos o un objeto de texto) y se utiliza la ventana propiedades o el menú contextual para dar formato al objeto.

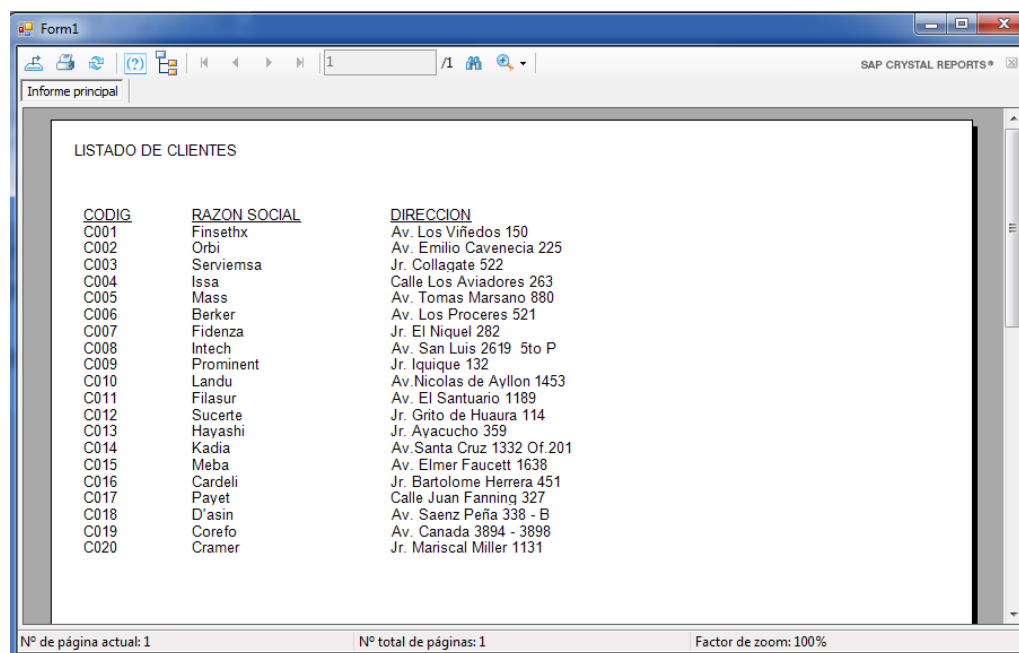
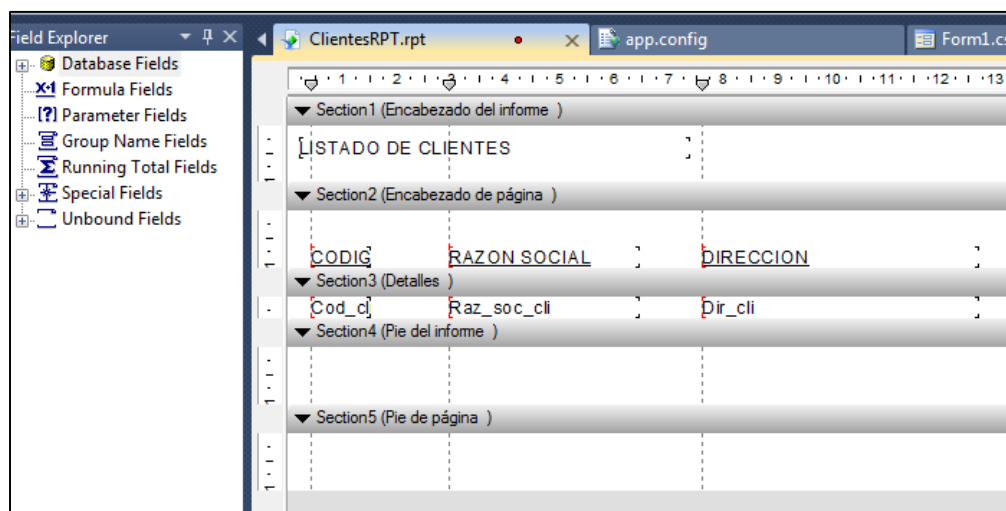
Algunos de los objetos que puede añadir al reporte y a los que puede dar formato según las necesidades, incluyen:

- Campos de base de datos
- Campos de fórmula
- Campos de parámetro
- Campos de nombre de grupo
- Campos de total acumulado
- Campos de resumen
- Gráficos
- Subreportes

Secciones de reportes

Crystal Reports está dividido en secciones de reporte, como los encabezados de las secciones, los pies de página y los detalles. Los objetos se arrastran a una sección del reporte.

Los datos que aparecen en el reporte finalizado dependen de las opciones de organización. En particular, los datos del reporte varían según las secciones en las que desee insertar objetos concretos. Por ejemplo, si inserta un objeto de gráfico en la sección encabezado, el gráfico solo aparecerá una vez al principio del reporte y resumirá los datos que contiene. Además, si un objeto de gráfico se añade a la sección encabezado de grupo, aparecerá un gráfico individual al principio de cada grupo de datos y solo se resumirán los datos relacionados con dicho grupo.



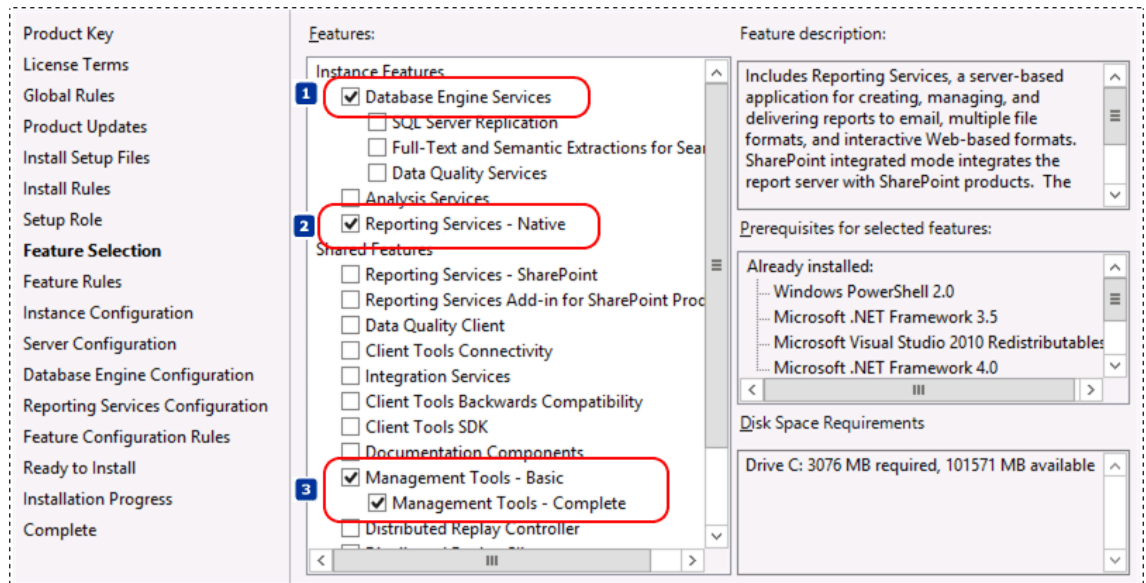
3. Reporting Services



Reporting Services es una plataforma de reportes basada en servidor, la misma que puede ser empleada para crear y administrar reportes tabulares, de matrices, gráficos y de libre formato, la información de estos reportes pueden provenir de diferentes orígenes de datos. Los reportes que se definen para ser administrados a través de una conexión basada en Web.

Reporting Services es parte de la suite de productos de SQL Server junto con Integration Services y Analysis Services. Durante el proceso de instalación de SQL Server se puede elegir el incluirlo y configurarlo.

Setup Role	
Click the SQL Server Feature Installation option to individually select which feature components to install, or click a feature role to install a specific configuration.	
<ul style="list-style-type: none"> Product Key License Terms Global Rules Product Updates Install Setup Files Install Rules Setup Role Feature Selection Feature Rules Feature Configuration Rules Ready to Install Installation Progress Complete 	<ul style="list-style-type: none"> <input checked="" type="radio"/> SQL Server Feature Installation Install SQL Server Database Engine Services, Analysis Services, Reporting Services, Integration Services, and other features. <input type="radio"/> SQL Server PowerPivot for SharePoint Install PowerPivot for SharePoint on a new or existing SharePoint server to support PowerPivot data access in the farm. Optionally, add the SQL Server relational database engine to use as the new farm's database server. <input checked="" type="checkbox"/> Add SQL Server Database Relational Engine Services to this installation. <input type="radio"/> All Features With Defaults Install all features using default values for the service accounts.



Ventajas de Reporting Services

- Una de las principales ventajas, es que se cuenta con una interface Web para lo que es la administración de los reportes. Desde esta interface se puede determinar en qué formato debe llegar el reporte, es decir, podemos decidir que el reporte llegue a una de las gerencias en formato PDF y para el departamento de consolidación y validación de datos podría enviarse la información en formato XML.
- Cuenta con un lenguaje de especificación estándar denominado Report Definition Language o simplemente RDL, el cual es un lenguaje de formato XML, que se encarga de definir el reporte.
- Con SQL Server Reporting Services, podemos conectarnos a cualquier repositorio de datos, a través de un .NET Data Provider, un proveedor OLE DB provider o uno de tipo ODBC.
- Para la distribución, los usuarios pueden acceder a los reportes en base a la infraestructura existente. Es decir, que los usuarios pueden acceder a los reportes a través de una barra de herramientas en el browser. Los reportes son accedidos desde un repositorio centralizado, presentado como un folder en orden jerárquico.
- Otra de las grandes características de Reporting Services, es que puede distribuir el reporte en distintos formatos, como hojas de Excel, documentos pdf, texto, XML, etc.
- La arquitectura de Reporting Services, permite a los desarrolladores preparar aplicaciones personalizadas que accedan a los reportes a través de una API que está expuesta como un Web service.

Controles ReportViewer en Visual Studio

Los controles ReportViewer son controles de ASP.NET y Windows Forms de Visual Studio que puede agregar a una página Web habilitada para AJAX en una aplicación o sitio Web ASP.NET y a un formulario en una aplicación de Windows Forms, respectivamente.

El control incrusta las funciones de visualización y procesamiento de reportes directamente en la aplicación. También, proporciona un entorno de diseño integrado para que pueda generar definiciones de reporte de cliente (.rdlc) utilizando datos de cualquier objeto de datos ADO.NET.

Una API que incorpore toda la funcionalidad ofrecerá acceso mediante programación al control y a los reportes para que pueda configurar la funcionalidad de reportes en tiempo de ejecución. Además, el control ASP.NET AJAX de Visual Studio introduce un conjunto de API de JavaScript, integrado en la biblioteca de AJAX, que permite a la aplicación host interactuar con el control ReportViewer, utilizando la programación del lado cliente.

4. Desarrollo de soluciones de Office

Desarrollo de soluciones Office

- Al usar Microsoft Office como frontend de nuestras soluciones, se pueden aprovechar las ventajas de las conocidas interfaces de usuario de Microsoft Office y herramientas tales como las funciones de procesamiento de textos de Word, las características de análisis de datos de Excel y las características de administración de correo electrónico de Outlook.

Al usar Microsoft Office como frontend de nuestras soluciones, se pueden aprovechar las ventajas de las conocidas interfaces de usuario de Microsoft Office y herramientas tales como las funciones de procesamiento de textos de Word, las características de análisis de datos de Excel y las características de administración de correo electrónico de Outlook.

Se puede desarrollar soluciones en Visual Studio para personalizar aplicaciones de Office y agregar las características concretas que necesita en procesos empresariales. Por ejemplo, se puede convertir Word en un generador de contratos que los ensamble a partir de elementos existentes, que pueden convertirse en editables o no editables.

Con Visual Studio podemos aprovechar las herramientas de diseño que nos proporciona el entorno de desarrollo de .NET y aunado al modelo de objetos de office y las clases del .NET Framework, desarrollar aplicaciones empresariales.

5. Tipos de soluciones Office



Visual Studio dispone de varios tipos de proyectos que admiten diferentes escenarios comerciales para Microsoft Office, entre los que se incluyen los siguientes tipos:

Tipos de proyecto	Aplicación de Microsoft Office
Personalizaciones de nivel de documento	Excel 2007 y Excel 2010 Word 2007 y Word 2010
Complementos de nivel de aplicación	Excel 2007 y Excel 2010 InfoPath 2007 e InfoPath 2010 Outlook 2007 y Outlook 2010 PowerPoint 2007 y PowerPoint 2010 Project 2007 y Project 2010 Visio 2007 y Visio 2010 Word 2007 y Word 2010

Los proyectos a nivel de documento se cargan al momento de abrir un documento y tienen un campo de acción sobre el documento en específico. Como ventaja tiene que la funcionalidad que se desarrolle acá estará cerrada solo a ese documento, si algún problema/excepción ocurre solo sucederá en el ámbito de ese documento. Como desventaja se presentaría el caso de excesivo consumo de memoria si se tienen abiertos varios documentos ya que se estaría cargando una instancia de nuestro proyecto para cada documento.

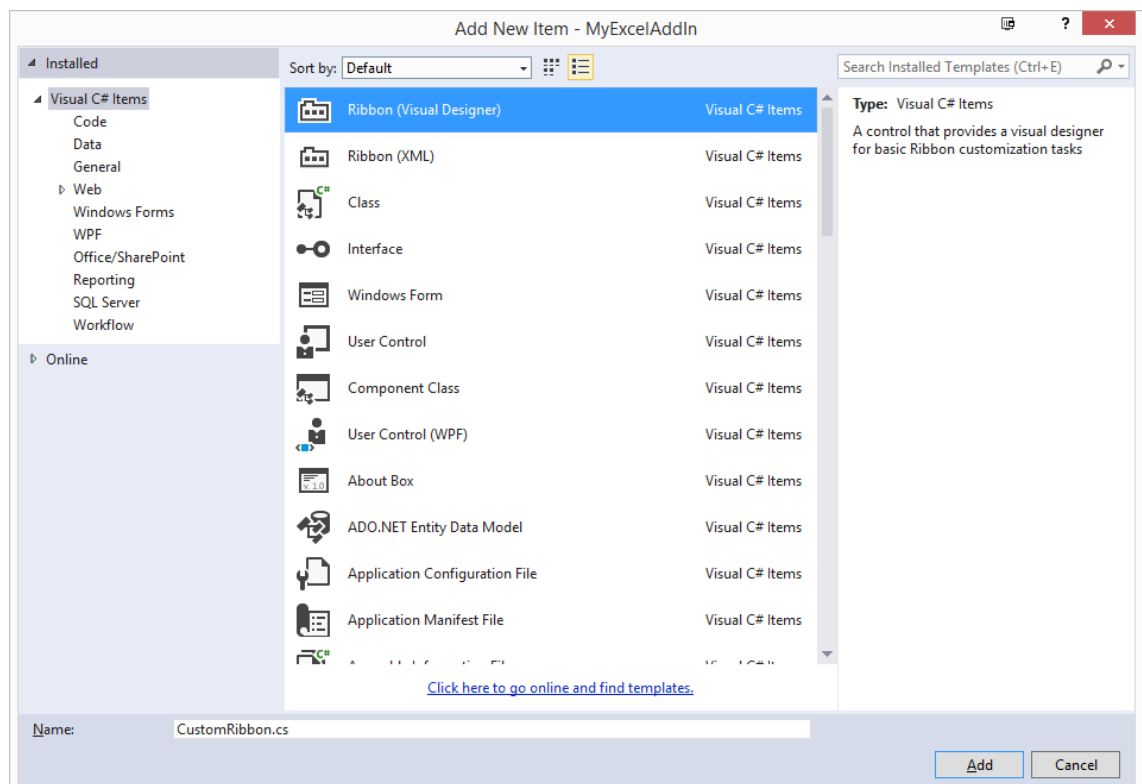
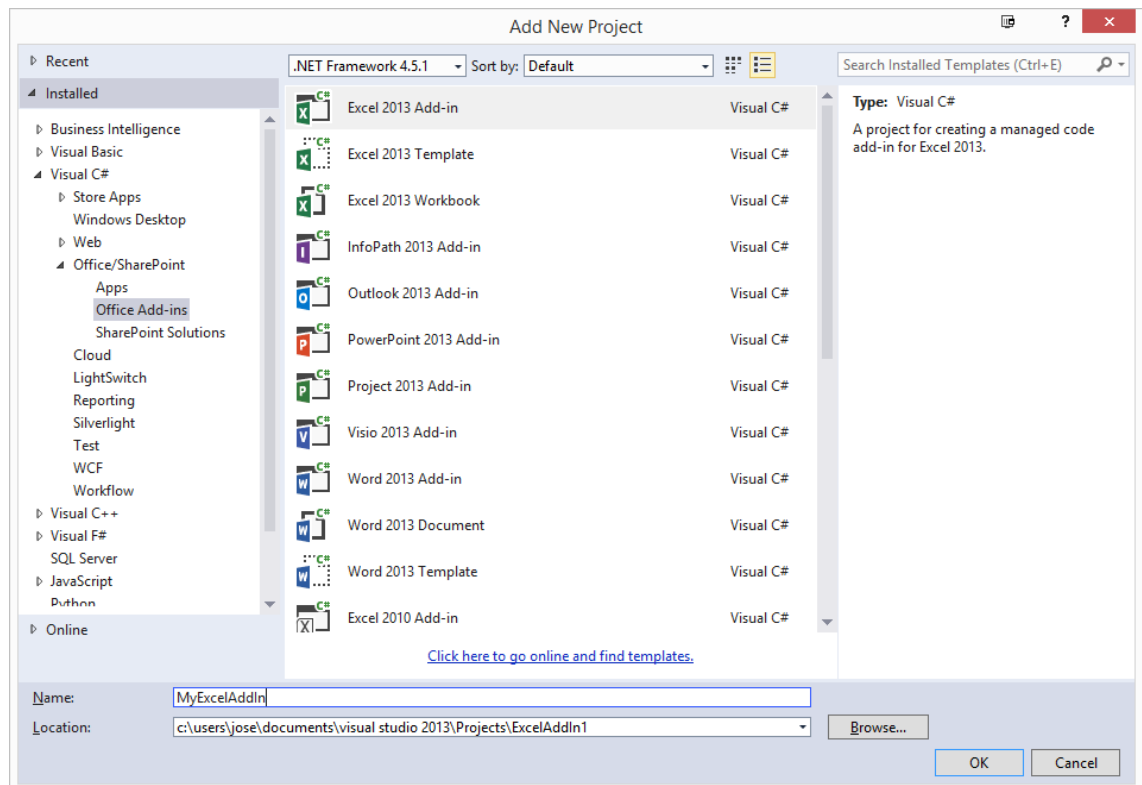
Los proyectos a nivel de aplicación se cargan al momento de iniciar la aplicación y su campo de acción puede abarcar varios documentos. La ventaja de estos proyectos es que es cargado una sola vez en nuestra aplicación, consumiendo menos recursos. Entre las desventajas que se tienen es que tenemos que asegurarnos de que nuestro proyecto soporte todas las versiones de documentos con las que se vaya a trabajar.

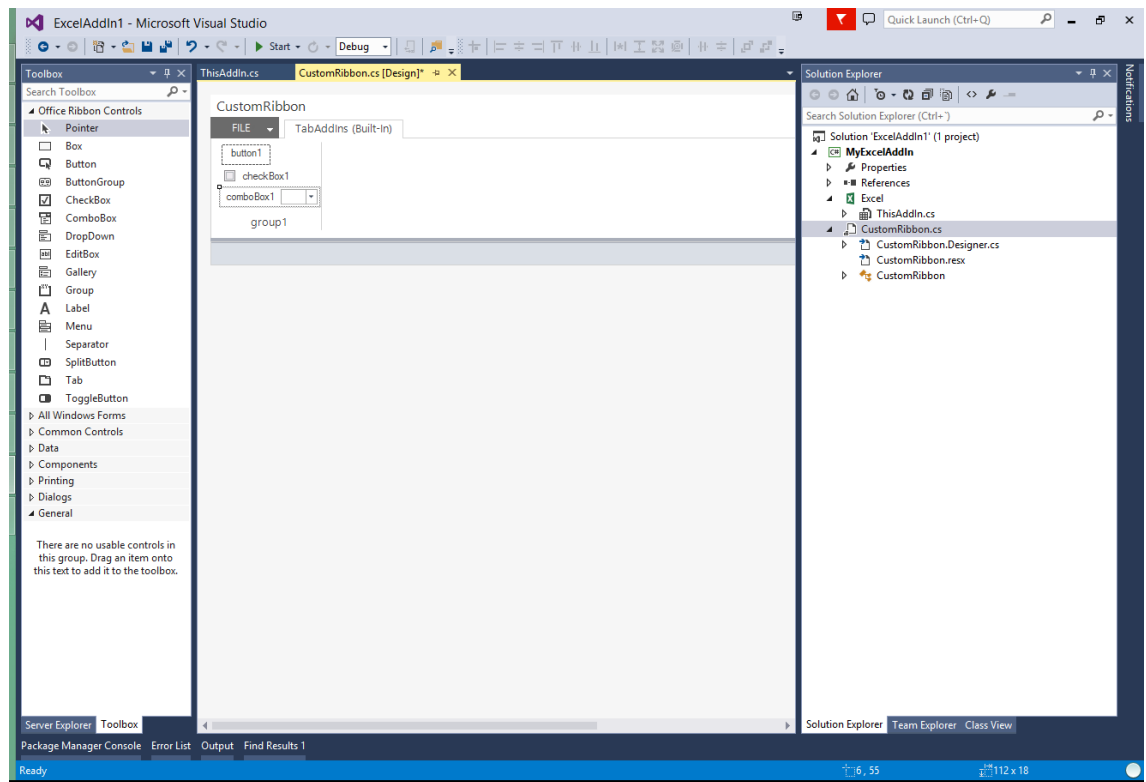
En la siguiente tabla, se muestran los tipos de proyectos que proporciona cada característica.

Característica	Tipos de proyectos que proporciona la característica
Panel de acciones.	Proyectos de nivel de documento.
Implementación de ClickOnce.	Proyectos de nivel de documento y de nivel de aplicación.
Paneles de tareas personalizados.	Proyectos de nivel de aplicación para las siguientes aplicaciones: <ul style="list-style-type: none"> • Excel • InfoPath • Outlook • PowerPoint • Word
Elementos XML personalizados.	Proyectos de nivel de documento. Proyectos de nivel de aplicación para las siguientes aplicaciones: <ul style="list-style-type: none"> • Excel • PowerPoint • Word
Caché de datos.	Proyectos de nivel de documento.
Exponer un objeto de una personalización en el nivel del documento en código de VBA en el documento.	Proyectos de nivel de documento.
Exponer un objeto de un complemento en el nivel de la aplicación en otras soluciones Microsoft Office.	Proyectos de nivel de aplicación.
Los controles host siguientes: <ul style="list-style-type: none"> • Chart • ListObject • NamedRange • Controles de contenido • Bookmark 	Proyectos de nivel de documento. Proyectos de nivel de aplicación para Word y Excel.

Los controles host siguientes: <ul style="list-style-type: none"> • XMLMappedRange • XMLNode • XMLNodes 	Proyectos de nivel de documento.
Implementación de varios proyectos.	Proyectos de nivel de documento. Proyectos de nivel de aplicación.
Áreas de formulario de Outlook.	Proyectos de nivel de aplicación para Outlook.
Acciones posteriores a la implementación.	Proyectos de nivel de documento. Proyectos de nivel de aplicación.
Personalizaciones de la cinta de opciones.	Proyectos de nivel de documento. Proyectos de nivel de aplicación para las siguientes aplicaciones: <ul style="list-style-type: none"> • Excel 2007 y Excel 2010 • InfoPath 2010 • Outlook 2007 y Outlook 2010 • PowerPoint 2007 y PowerPoint 2010 • Project 2010 • Visio 2010 • Word 2007 y Word 2010
Etiquetas inteligentes (con el modelo de programación para los proyectos de Office en Visual Studio, en lugar del SDK de etiquetas inteligentes).	Proyectos de nivel de documento. Proyectos de nivel de aplicación para Word y Excel.
Diseñador visual de documentos.	Proyectos de nivel de documento.

Dentro de las capacidades que tienen las aplicaciones Office se tiene la creación de cintas de opciones (Ribbon Visual Designer). Esta cinta de opciones es el menú característico que presentan las aplicaciones Office desde su versión 2007. Con las aplicaciones Office podemos crear nuevos menús de una manera familiar (estilo de trabajo similar al de Winforms, que permite arrastrar y soltar controles que deseamos incluir). Para crear este tipo de cinta de opciones se tiene que crear un proyecto de tipo AddIn (por ejemplo ExcelAddIn) y una vez que se tenga creado el proyecto agregar un Item (Add > New Item) de tipo Ribbon.





Para probar las aplicaciones se tiene que seleccionar una versión de Office que tengamos instalada. Para poder cambiar la versión por defecto que se tiene seleccionada en nuestro proyecto, debemos abrir la carpeta de nuestro proyecto y abrir el archivo de nuestro proyecto (extensión .csproj o .vbproj en un editor de texto)

GitRepos	obj	12/04/2015 08:35 ...	Carpeta de archivos	
GitRepos2	Properties	12/04/2015 08:35 ...	Carpeta de archivos	
Graphics	CustomRibbon	12/04/2015 08:37 ...	Archivo CS	1 KB
IISExpress	CustomRibbon.Designer	12/04/2015 08:37 ...	Archivo CS	3 KB
Instaladores	CustomRibbon	12/04/2015 08:38NET Managed Re...	6 KB
JRiver	MyExcelAddIn	12/04/2015 08:35 ...	Archivo CSPROJ	10 KB
My Web Sites	ThisAddIn	12/04/2015 08:35 ...	Archivo CS	1 KB
Plantillas perso	ThisAddIn.Designer	12/04/2015 08:35 ...	Archivo CS	12 KB
Projects	ThisAddIn.Designer	12/04/2015 08:35 ...	Archivo XML	2 KB
SQL Server Mar				
Visual Studio 20				
Visual Studio 20				
Visual Studio 20				
ArchitectureE				
Backup Files				

En el editor de texto debemos buscar el valor de `DebugInfoExeName` y cambiar dicho valor a la versión de Office con la que deseemos trabajar.

```

MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(
VisualStudioVersion)\VSToolsPath>
194 </PropertyGroup>
195 <!-- Include the build rules for a C# project. -->
196 <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets
" />
197 <!-- Include additional build rules for an Office
application add-in. -->
198 <Import Project="$(VSToolsPath)\OfficeTools\Microsoft.
VisualStudio.Tools.Office.targets" Condition="'$(
VSToolsPath)' != ''" />
199 <!-- This section defines VSTO properties that describe the
host-changeable project properties. -->
200 <ProjectExtensions>
201 <VisualStudio>
202 <FlavorProperties GUID="{BAA0C2D2-18E2-41B9-852F-
F413020CAA33}">
203 <ProjectProperties HostName="Excel" HostPackage="
{29A7B9D7-A7F1-4328-8EF0-6B2D1A56B2C1}" OfficeVersion="
15.0" VstxVersion="4.0" ApplicationType="Excel"
Language="cs" TemplatesPath="" DebugInfoExeName="
#Software\Microsoft\Office\15.
0\Excel\InstallRoot\Path#excel.exe"
DebugInfoCommandLine="/x" AddItemTemplatesGuid="
{51063C3A-E220-4D12-8922-BDA915ACD783}" />
204 <Host Name="Excel" GeneratedCodeNamespace="MyExcelAddIn
" IconIndex="0">
205 <HostItem Name="ThisAddIn" Code="ThisAddIn.cs"
CanonicalName="ThisAddIn" CanActivate="false"
IconIndex="1" Blueprint="ThisAddIn.Designer.xml"
GeneratedCode="ThisAddIn.Designer.cs" />
206 </Host>
207 </FlavorProperties>
208 </VisualStudio>
209 </ProjectExtensions>
210 </Project>

```

Valor para Excel 2013

`DebugInfoExeName="#Software\Microsoft\Office\15.0\Excel\InstallRoot\Path#excel.exe"`

Valor para Excel 2010

`DebugInfoExeName="#Software\Microsoft\Office\14.0\Excel\InstallRoot\Path#excel.exe"`

Valor para Excel 2007

`DebugInfoExeName="#Software\Microsoft\Office\12.0\Excel\InstallRoot\Path#excel.exe"`

Otra de las capacidades que presentan las aplicaciones Office es la modificación del DOM (Document Object Model), es decir que mediante programación se puede modificar la estructura de un documento (Excel, Word, etc.) cambiando valores de celdas, de palabras, creando nuevas hojas, etc. Para poder realizar estos cambios uno debe familiarizarse con el API que permite el acceso al DOM de los documentos.

Word DOM



Excel DOM



Por ejemplo, podemos realizar búsqueda de valores en hojas de Excel:

```
internal void SearchUsingDOM(string searchcriteria)
{
    //Aca podemos llamar a BD, WS, etc etc
    var nombresArray = new[] { "Jose", "Pedro", "Carlos" };
    var workSheet = this.Application.ActiveWorkbook.ActiveSheet;
    var i = 0;
    foreach (var item in nombresArray)
    {
        i++;
        string rango = string.Format("A{0}:A{0}", i);
        var row = workSheet.Range[rango];
        row[1, 1] = item;
    }
}
```