

Capítulo 11

Single Page Application (SPA) con React

Objetivo

Al finalizar el capítulo, el alumno:

- Implementar Frontend con ReactJs.
- Utilizar los componentes ReactJS.
- Administrar estados con Redux.

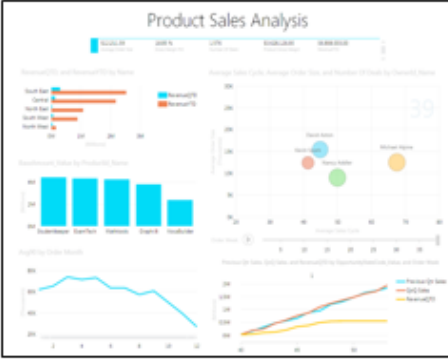
Temas

1. Introducción a SQL Server Reporting Services.
2. Construyendo un Reporte en SSRS.
3. Trabajando con Reportes en ASP.NET MVC.

1. Introducción a SQL Server Reporting Services

Introducción a SQL Server Reporting Services

- Reporting Services es una plataforma de reportes basada en SQL Server.
- Proporciona funcionalidad completa de generación de reportes para una gran variedad de orígenes de datos.



11 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

SQL Server Reporting Services dispone de una gama completa de herramientas y servicios que ayudan crear, implementar y administrar reportes, así como, también ofrece características de programación que permitirán extender y personalizar la funcionalidad de los reportes.

Es una plataforma de reportes que proporciona funcionalidad completa para una gran variedad de orígenes de datos. Además, las API permiten a los desarrolladores integrar o ampliar el procesamiento de datos o reportes en aplicaciones personalizadas. Las herramientas de Reporting Services funcionan en el entorno de Visual Studio y están totalmente integradas con las herramientas y componentes de SQL Server.

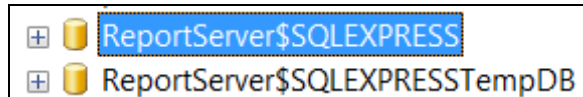
Se pueden crear reportes interactivos, tabulares, gráficos, entre otros, a partir de orígenes de datos relacionales, multidimensionales o basados en XML.

Los reportes pueden incluir visualización de datos avanzada, como diagramas, mapas y gráficos. Además, se puede publicar, programar su procesamiento o acceder a ellos a demanda. También, se puede elegir entre varios formatos de visualización, exportarlos a otras aplicaciones como Microsoft Excel y suscribirse a los reportes publicados. Los reportes creados se pueden ver mediante una conexión basada en web o como parte de una aplicación de Microsoft Windows o un sitio de SharePoint. Puede crear también alertas de datos en los reportes publicados en un sitio de SharePoint y recibir mensajes de correo electrónico cuando cambien los datos.

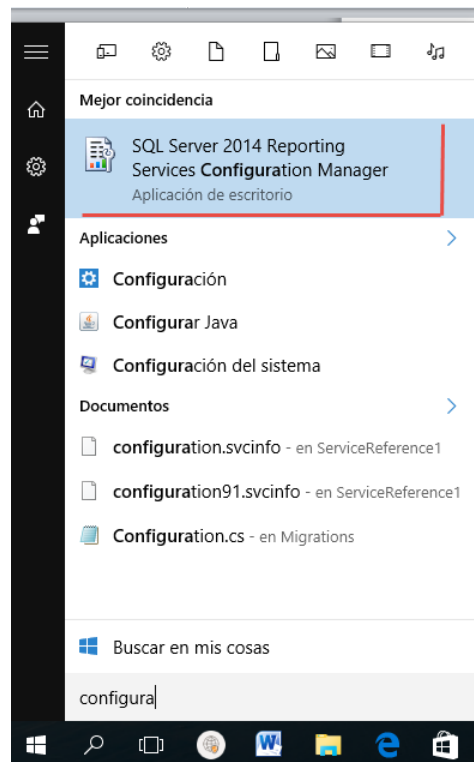
Para saber que tenemos instalado y funcionando correctamente Reporting Services, se debe verificar que se esté ejecutando el servicio "SQL Server Reporting Services", en caso se use la versión SQL Express, aparecerá esto entre paréntesis como en la imagen siguiente:

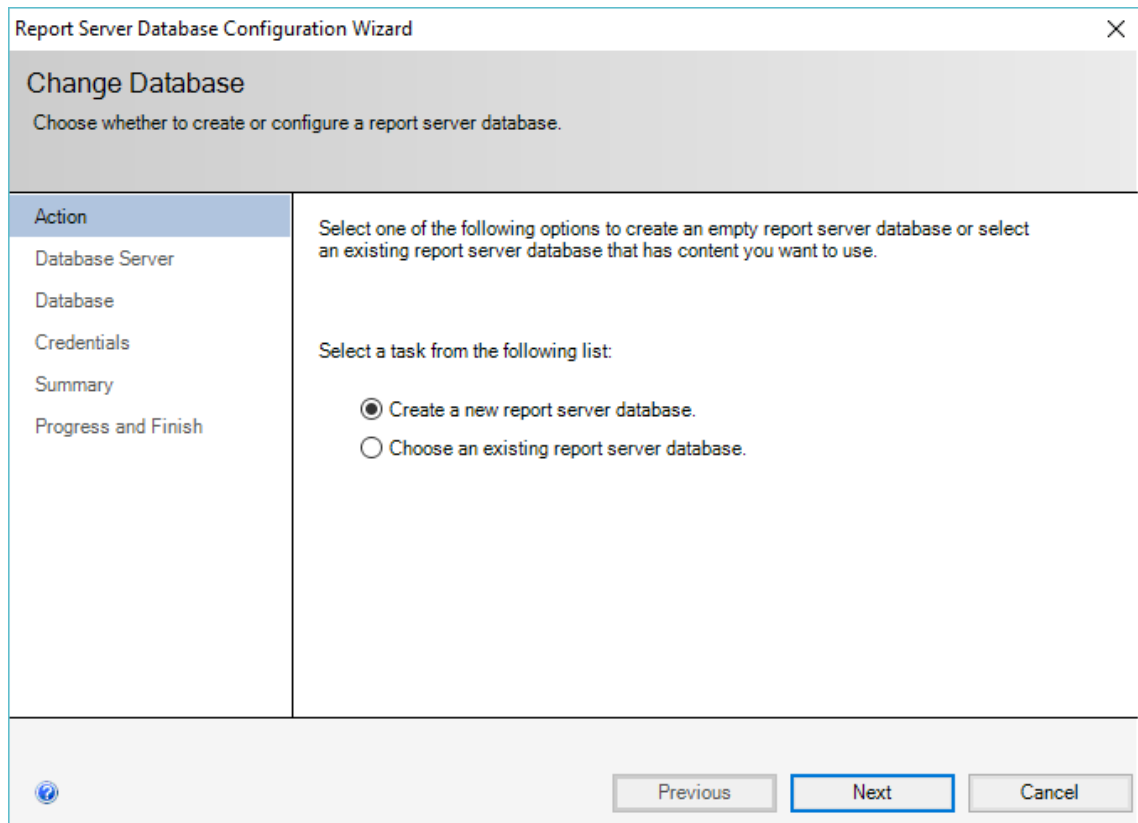
SQL Server VSS Writer	Provides the interface to backup/restore Microsoft SQL server throug...	Running	Automatic	Local System
SQL Server Reporting Services (SQLEXPRESS)	Manages, executes, renders, schedules and delivers reports.	Running	Automatic	NT Service\ReportServer\$SQLEXPRESS
SQL Server Browser	Provides SQL Server connection information to client computers.	Disabled		Local Service
SQL Server Agent (SQLEXPRESS)	Executes jobs, monitors SQL Server, fires alerts, and allows automatio...	Disabled		Network Service
SQL Server (SQLEXPRESS)	Provides storage, processing and controlled access of data, and rapi...	Running	Automatic	NT Service\MSSQL\$SQLEXPRESS
SQL Full-text Filter Daemon Launcher (SQLEXPRESS)	Service to launch full-text filter daemon process which will perform d...	Running	Manual	NT Service\MSSQLFDLauncher\$SQLEXPRESS

Luego, verificar que se tengan las bases de datos del Report Server:



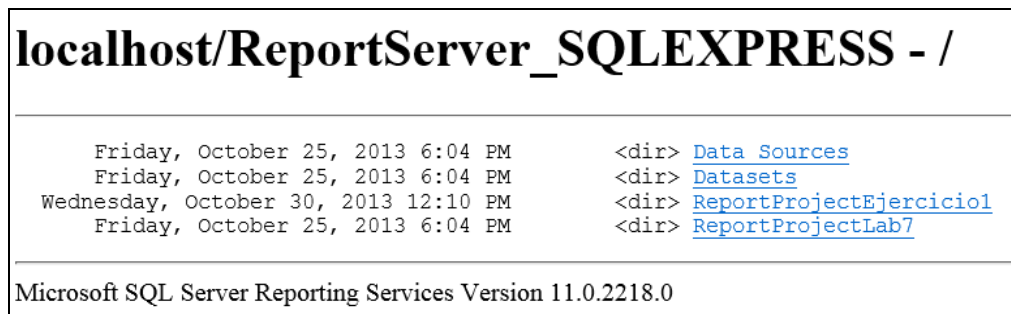
En caso no se cuenta con la base de datos instalada se debe proceder con su instalación en SQL Server <Version> Reporting Services Configuration Manager:



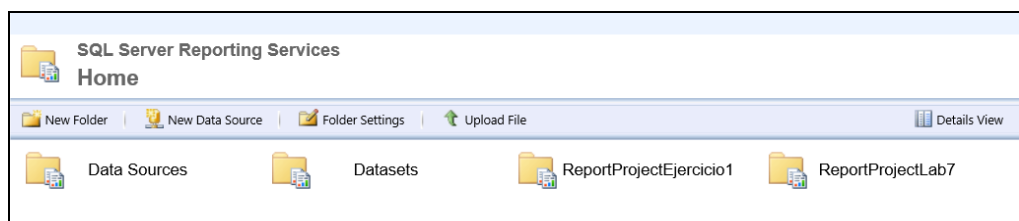


Finalmente, se puede acceder al visor de reportes usando IE como administrador de cualquiera de las 2 formas siguientes: (los reportes que se listan, se deben haber construido previamente, tal como se verá en la siguiente sección).

- http://localhost/reportserver_sqlexpress: Url del servicio web.



- http://localhost/reports_sqlexpress: Url del administrador de reportes.



2. Construyendo un Reporte en SSRS

Construyendo un Reporte en SSRS

- Para construir Reportes se hace a través de la herramienta SQL Server Data Tools (SSDT), ejecutándola como administrador. Y luego se procede a publicarlo en el servidor de SQL Server Reporting Services (SSRS).
- Descargar el SSDT: <https://www.microsoft.com/es-es/download/details.aspx?id=42313>

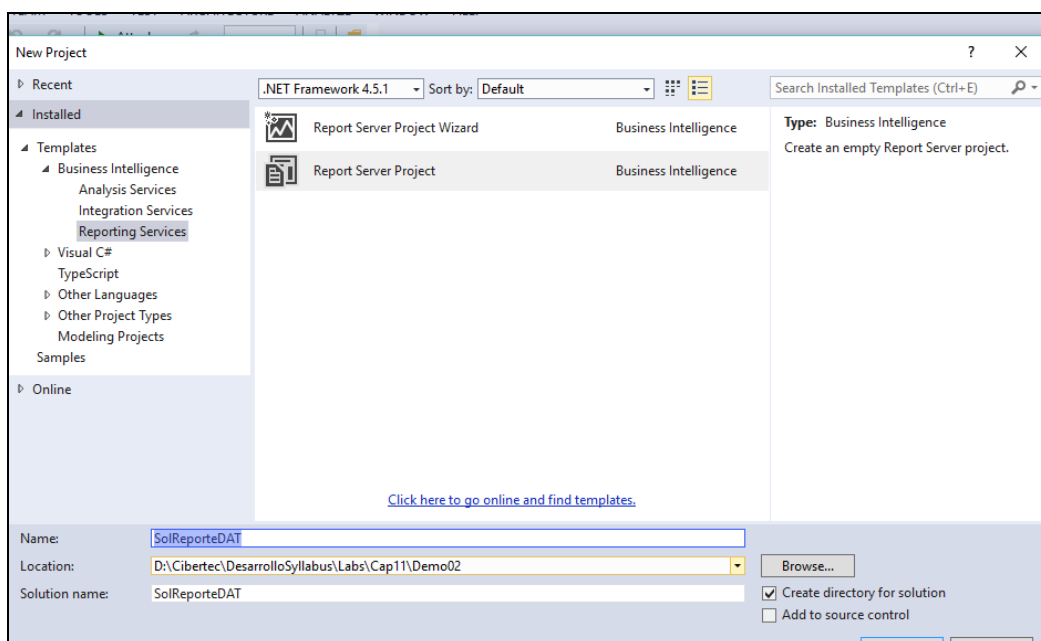


11 - 9 Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

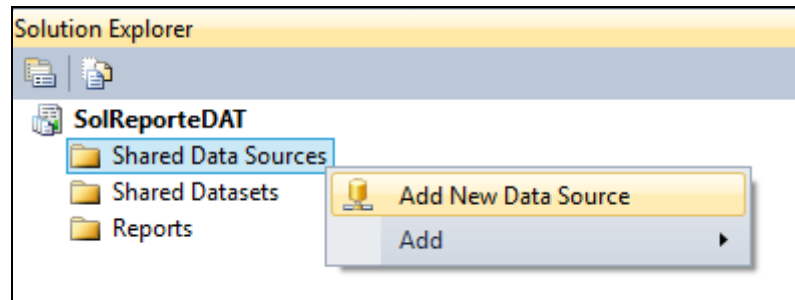
Para crear reportes en SQL Server Reporting Services, se debe usar SQL Server Data Tools (el cual debe ser ejecutado con permisos de administrador).

Pasos para construir un reporte:

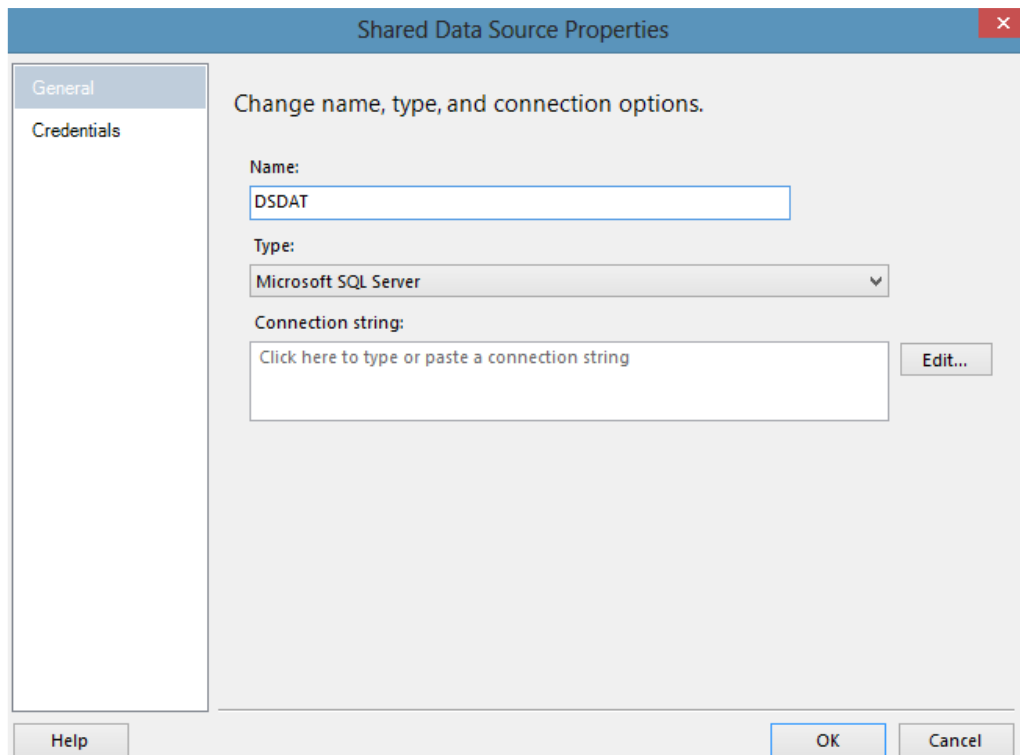
1. Crear un nuevo proyecto, para lo cual se debe seleccionar la opción "Report Server Project", que se encuentra en la sección Reporting Services, y colocarle el nombre y ruta donde se desee almacenar la solución.



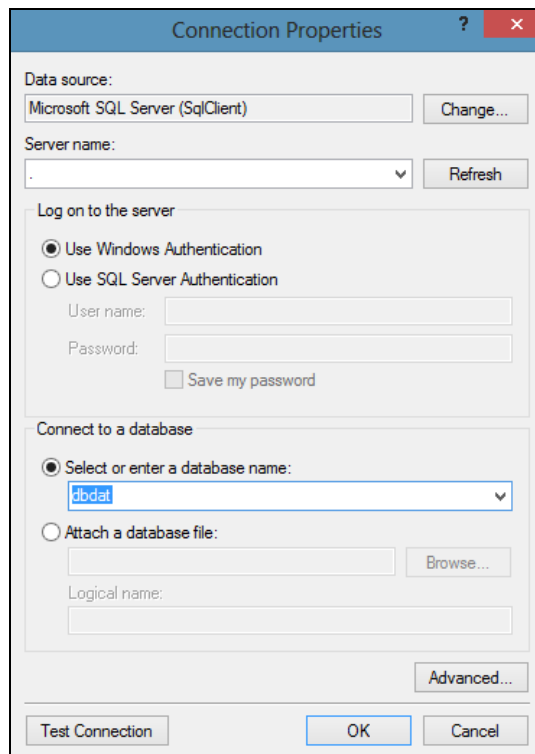
2. Se procede a crear un origen de datos, para lo cual se debe hacer clic derecho sobre Shared Data Sources y luego en Add New Data Source: (la opción de shared data sources solo funciona en la versión completa de SQL Server).



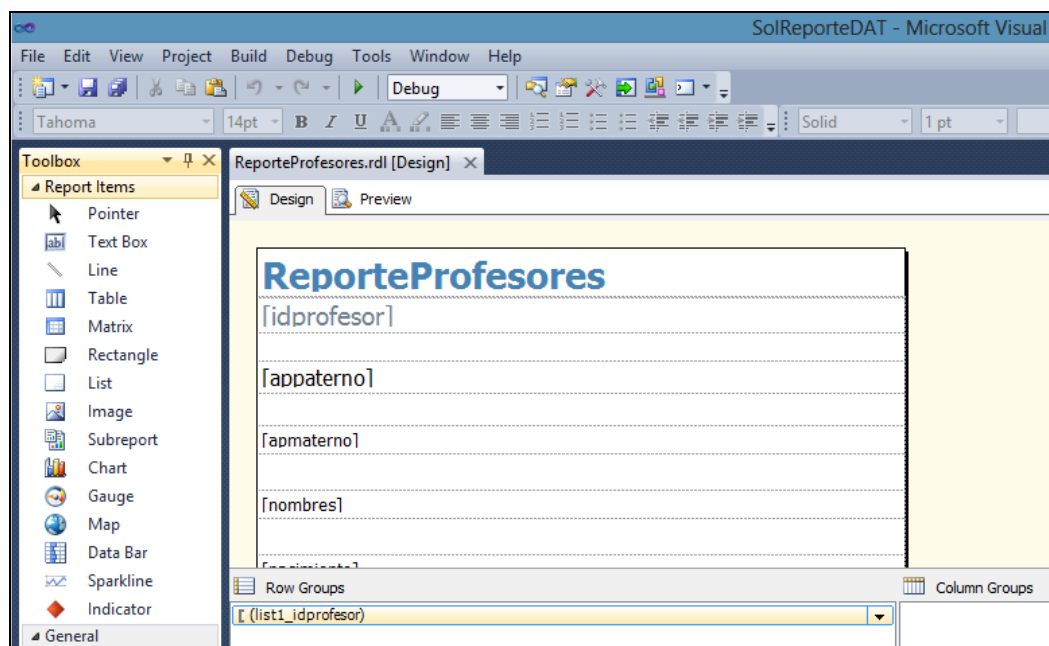
3. Se debe ingresar el nombre, tipo y cadena de conexión para la información del Data Source. En este caso, se selecciona la opción Credentials (se encuentra en el panel izquierdo) para establecer la configuración al servidor de base de datos.



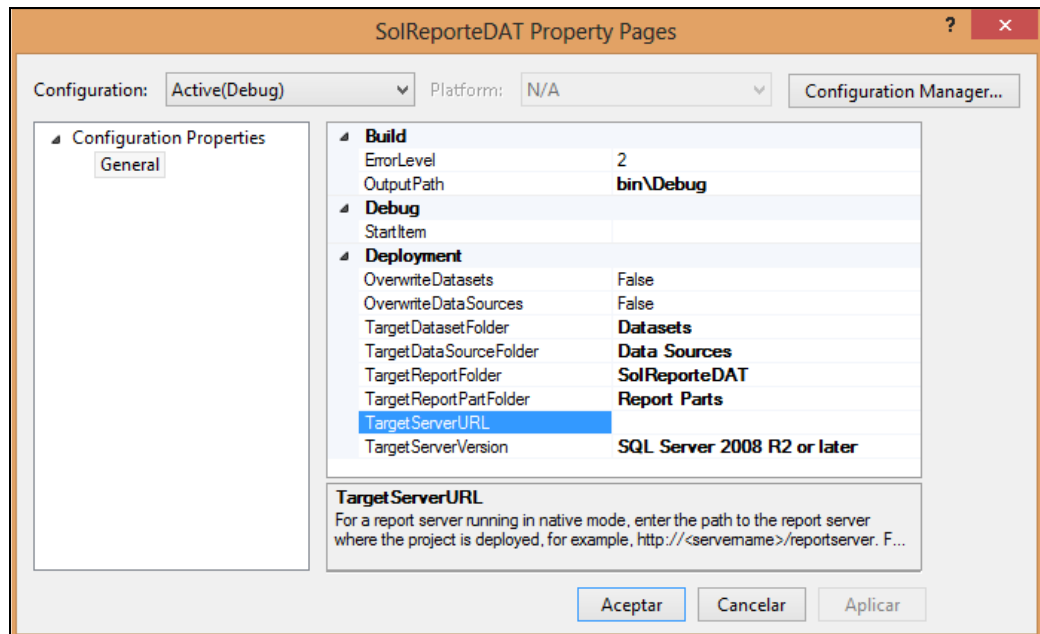
- Se selecciona el servidor de base de datos, tipo de autenticación y base de datos. Validar con Test Connection y, finalmente, hacer clic en OK.



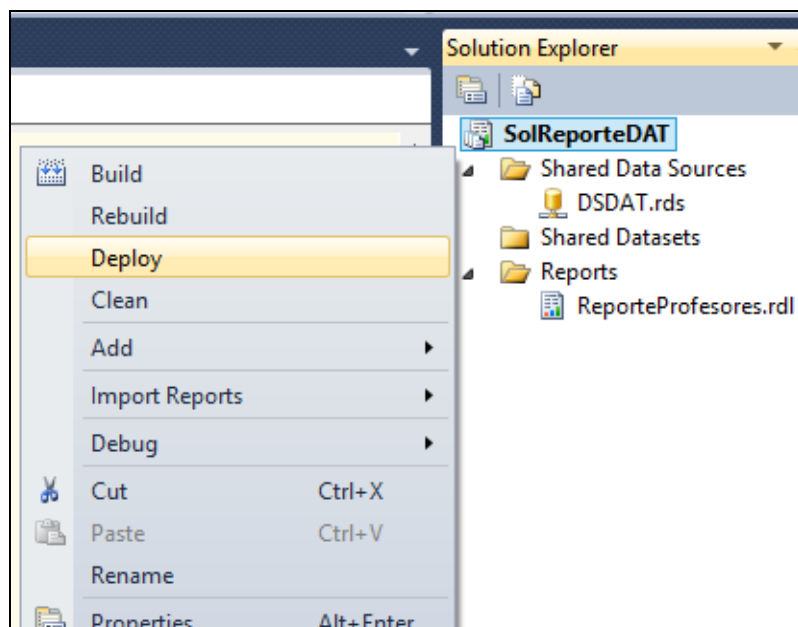
- En la vista de Editor de Reporting, se puede construir y personalizar el reporte según las características que se desee. Además, se dispone de un panel de herramientas con controles básicos, pero además se pueden importar nuevos controles para casos más específicos.



6. Es necesario establecer la configuración adecuada, para lo cual al hacer clic derecho sobre el proyecto de reportes, se debe seleccionar la opción Propiedades. En esta ventana, se debe completar la información de la propiedad TargetServerURL, en la que se especifica el URL en donde se hará deploy del reporte. (Formato: <http://nombreservidor/reportserver>)



7. Finalmente, se debe seleccionar la opción Deploy para iniciar el proceso de publicación.




3. Trabajando con Reportes en ASP.NET MVC

Trabajando con Reportes en ASP.NET MVC

- Para trabajar con Reportes en ASP.NET MVC existen dos formas:
 - Renderizar el contenido del Reporte en Formato HTML sobre una vista.
 - Serializar a JSON el contenido del Reporte en Formato HTML y renderizarlo a través de Ajax.
- En ambos casos, la petición hacia el servidor de SSRS será a través del atributo src definido en un iframe, que contendrá al reporte y que debe ser construido dinámicamente en el servidor.

11 - 18

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.



En las aplicaciones ASP.NET Web Forms existe un control llamado ReportViewer, que representa un visor de reporte, en el cual se pueden establecer las propiedades para enlazarse con un servidor de reportes. Pero en el caso de ASP.NET MVC no existe este control ya que es una arquitectura y forma de trabajo diferente.

Para trabajar en MVC con un reporte previamente creado de Reporting Services, en primer lugar, se debe crear un método en el controller que permita construir el reporte, para lo cual se tiene que construir dinámicamente un HTML que tenga como salida un iframe. Dicho iframe debe tener como propiedad el atributo src, que se debe armar según la documentación oficial de parámetros de acceso URL para la carga de reportes. Existen 3 tipos de parámetros de acceso URL que se pueden establecer y que están detallados en el siguiente enlace:

<http://technet.microsoft.com/es-ve/library/ms152835.aspx>

- Comandos del Visor HTML (rc:)
- Comandos del servidor de informes (rs:)
- Comandos de elemento web del visor de informes (rv:)

Por cada tipo de Comandos, existe un listado de parámetros disponibles para utilizar en la construcción del URL. En este caso se usarán 3 parámetros:

- Parameters: Muestra u oculta el área de parámetros de la barra de herramientas. Si se establece este parámetro en el valor true, se muestra el área de parámetros de la barra de herramientas, pero si se establece este parámetro en false, el área de parámetros no se muestra al usuario. El valor por default de este parámetro es true.
- Command: Realiza una acción en un elemento del reporte, según el tipo de elemento. El valor predeterminado se determina mediante el tipo de elemento del reporte al que se hace referencia en la cadena de acceso URL. Los valores válidos son:

- ListChildren: muestra los elementos secundarios del elemento pasado a la dirección URL. Los elementos secundarios se muestran dentro de una página genérica de navegación por elementos.
- Render: representa el reporte especificado.
 - GetSharedDatasetDefinition: muestra la definición XML asociada a un conjunto de datos compartido. Para usar este valor, se debe tener el permiso para leer definición de reporte en un shared data set.
 - GetDataSourceContents: muestra las propiedades de un origen de datos compartido determinado. Se muestran las propiedades como XML.
 - GetResourceContents: representa un recurso y se muestra en una página HTML.
 - GetComponentDefinition: muestra la definición XML asociada a un elemento de reporte publicado. Para usar este valor, se debe tener el permiso de lectura contenido en un elemento del reporte publicado.
 - Format: Especifica el formato en el que se representará un reporte. Los valores habituales son ATOM, HTML4.0, MHTML, IMAGE, EXCEL, WORD, CSV, PDF, XML. El valor predeterminado es HTML4.0.

Con esta información, se crea el siguiente action method:

```
public string ConstruirReporte(string id)
{
    //URL del Servidor de Reporting Services
    string servidor = "http://localhost/ReportServer SQLEXPRESS";

    //Carpeta donde tenemos los reportes
    string carpeta = "ReportProjectEjercicio1";

    //Nombre del Reporte
    string reporte = "ReporteOrdenesPorCliente";

    //Los parámetros con sus respectivos valores
    string parametros = "&CustomerID=" + id.Trim();

    //Comandos a pasar al Visor de Reporting Services
    //http://technet.microsoft.com/es-ve/library/ms152835.aspx
    string comandosRS = "&rs:Command=Render&rs:Format=HTML4.0&rc:Parameters=false";

    return String.Format("<iframe id='ifReporte' width='100%' style='height: 480px'" +
        "frameborder='0' src='{0}/{1}/{2}/{3}/{4}'></iframe>",
        servidor, carpeta, reporte, parametros, comandosRS);
}
```

Luego, hay 2 formas en que se puede mostrar el reporte, síncrono y asíncrono:

1. Síncrono

- Se crea un action method que envíe el reporte en el ViewBag:

```
public ActionResult Reporte(string id)
{
    ViewBag.ReporteHTML = ConstruirReporte(id);
    return View();
}
```

- Se crea una vista que muestre el contenido del campo del ViewBag que tiene al reporte enviado desde el controller:

```
@{
    ViewBag.Title = "Reporte";
}

<h2>Reporte</h2>

@Html.Raw(HttpUtility.HtmlDecode(ViewBag.ReporteHTML.ToString()))
```

2. Asíncrono

- Se crea un action method que retorne reporte en formato JSON:

```
public JsonResult ReporteAsincrono(string id)
{
    return Json(ConstruirReporte(id), JsonRequestBehavior.AllowGet);
}
```

- Se crea una vista que permita cargar el reporte dentro de algún contenedor (en el ejemplo, tenemos al span con id spnReporte) y se agrega la funcionalidad de ajax con jQuery para cargar el reporte.

```
@{
    ViewBag.Title = "ReporteAsincrono";
}

<h2>Reporte de Ordenes por Cliente</h2>

Seleccione el cliente:
@Html.DropDownList("clientes", (IEnumerable<SelectListItem>)ViewBag.Clientes,
    new { style = "max-width: 400px;" })

<input type="button" value="Ver Reporte" id="btnVerReporte" />

<br />

<span id="spnReporte"></span>

@section Scripts {
    <script type="text/javascript">
        $(function () {
            $('#btnVerReporte').click(function () {
                $.getJSON(
                    '/Home/ReporteAsincrono',
                    { id: $('#clientes').val() },
                    function (result) {
                        $('#spnReporte').html(result);
                    });
            });
        });
    </script>
}
```