

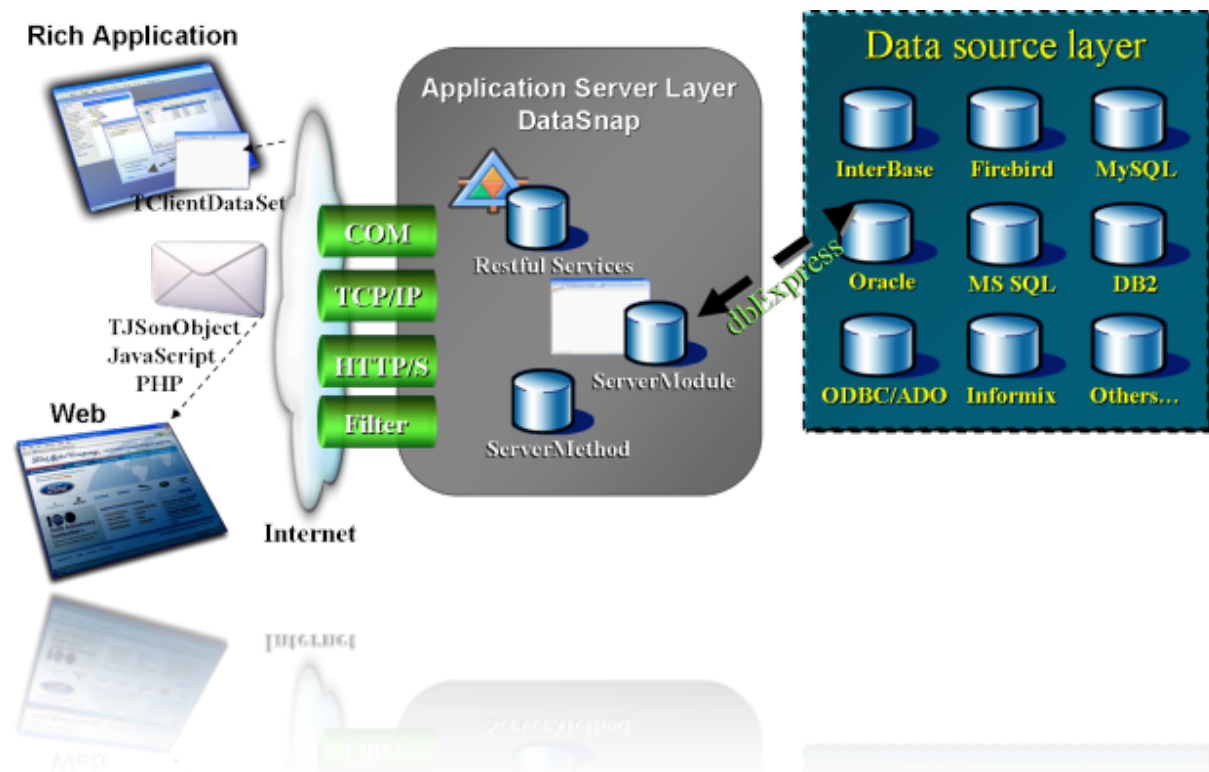
Contenido del Material

Introducción.....	2
Descripción y Arquitectura de DataSnap.....	3
Descripción:	3
Arquitectura:.....	4
Ejemplo de DataSnap Simple (Como en Primero).....	5
Servidores DataSnap que podemos crear con los Ayudantes	7
DataSnap Server Wizard:.....	8
DataSnap REST Application Wizard:.....	14
DataSnap WebBroker Application Wizard:.....	19
Componentes de la paleta “DataSnap Server”	25
Ejemplos de Servidores y Clientes DataSnap usando los Asistentes	27

Introducción

La tecnología DataSnap sigue evolucionando, por el aumento en la demanda de aplicación distribuidas. La tecnología detrás de DataSnap se ha movido del enfoque remoto soportado por la plataforma Microsoft COM/DCOM a un enfoque de comunicación más abierta basada en TCP/IP. Esta evolución ha permitido que la tecnología DataSnap pudiese ampliar sus capacidades con la finalidad de soportar una capa intermedia completa. Una de las principales características de la tecnología es que es mucho más rápido: rápido para desarrollar, rápido de implementar y rápido y fácil de ejecutar o colocar en la producción.

La ampliación de las capacidades de DataSnap, nos permite desarrollar aplicaciones basadas en estándares. DataSnap por compatibilidad mantiene el soporte a COM/DCOM, además ahora tiene la capacidad de comunicarse de forma nativa a través de TCP/IP, alternativamente, a través de HTTP o HTTPS. Al mismo tiempo, la lógica del negocio que se encuentra en los servidores DataSnap, pueden ser despachados a los clientes como servicios RESTful.



Descripción y Arquitectura de DataSnap

Descripción:

Anteriormente conocido como MIDAS, DataSnap Delphi es una tecnología que permite el desarrollo de aplicaciones multi-capas, especialmente aplicaciones multi-capas de base de datos. También hay soporte completo para C++ Builder. DataSnap es independiente de MIDAS. DataSnap ofrece la posibilidad de crear aplicaciones cliente-servidor que se comunican a través de Internet, redes locales y servidores dedicados.

La característica principal de DataSnap es la capacidad que tiene la aplicación cliente para invocar métodos que se implementan en un servidor. DataSnap genera automáticamente las interfaces necesarias, para que el cliente pueda comunicarse con el servidor que contiene los prototipos de los métodos remotos.



DataSnap proporciona una forma para que el Cliente pueda comunicarse de forma segura con el servidor, mediante una transferencia de datos de manera segura en formato JSON (JavaScript Object Notation) a través de TCP/IP o HTTP(S). Permite definir filtros en ambos extremos del canal de comunicación, entre ellos: cifrado y

compresión de la data transferencia, con el propósito de mejora la seguridad y el rendimiento en la transferencia, además permite crear nuestros propios filtros.

Otro de los beneficios de la tecnología DataSnap es que ofrece la posibilidad de notificar de forma asíncrona a todas las aplicaciones clientes conectadas de los cambios realizados en el servidor, de modo que los clientes puedan tomar las acciones apropiadas, ésto se logra a través de la implementación de “callbacks”.

Arquitectura:

La tecnología DataSnap permite el desarrollo de aplicaciones cliente-servidor, donde el cliente o el servidor pueden ser desarrollados en Delphi o C++Builder. La comunicación entre el cliente y el servidor se pueden hacer en el mismo equipo, en una red local, o a través de la Internet. Usted puede elegir el tipo de conexión estableciendo la propiedad `CommunicationProtocol` de un componente `TSQLConnection` a TCP/IP o HTTP(S), en tiempo de diseño.

La comunicación entre cliente y servidor se realiza a través del intercambio de contenidos de datos JSON (JavaScript Object Notation). La respuesta del servidor también se pueden enviar en formato HTML, utilizando la tecnología WebBroker.

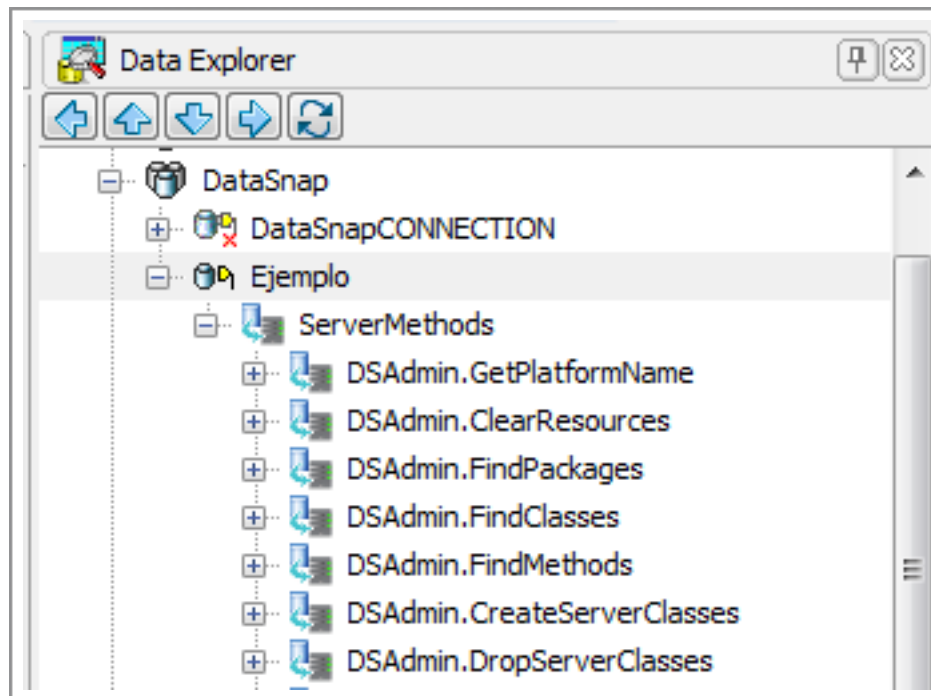
Si un cortafuegos está en ejecución en el cliente o en el servidor, DataSnap ofrece la opción de enrutar la transferencia de datos a través de un túnel, anulando el cortafuegos, permitiendo de esta manera la comunicación entre el cliente y el servidor.

DataSnap permite la implementación del cliente y del servidor en la misma aplicación, también conocida como conectividad “in-process”.

Visualizando de Métodos Remotos usando “DataExplorer”:

Para utilizar DataExplorer con DataSnap, primero es necesario tener un servidor de aplicaciones DataSnap ejecutándose. Una vez que esta aplicación de servidor está en funcionamiento, vaya a la pestaña DataExplorer en la interfaz principal de Delphi.

En la ventana DataExplorer, en la sección DataSnap, nos damos cuenta de que podemos visualizar la categoría: “ServerMethods”, tal como se muestra en la imagen siguiente:



Ejemplo de DataSnap Simple (Como en Primero)

Principales Componentes del Servidor DataSnap:

- [TDSServer](#): El componente [TDSServer](#) es el corazón lógico del servidor DataSnap. Contiene métodos [Start](#) y [Stop](#) para iniciar y detener el servidor. También contiene la propiedad [AutoStart](#). Por defecto el valor de [AutoStart](#) es `True`, por lo que el servidor se inicia automáticamente cuando la aplicación se ejecuta. Solo hay un componente [TDSServer](#) por aplicación de servidor.
- [TDSServerClass](#): representa una clase de servidor. Al igual que [TDSServer](#) es necesario un componente [TDSServerClass](#) para proveer la comunicación entre el servidor y el cliente. El servidor DataSnap automáticamente crea y destruye instancias de clases del servidor. La creación de instancias de una clase de servidor se controla mediante la propiedad [LifeCycle](#) del componente [TDSServerClass](#). La propiedad [LifeCycle](#) tiene tres valores posibles: [Server](#), [Session](#), e [Invocación](#).
 - [Server](#) significa que el Servidor DataSnap crea una sola instancia de la clase del servidor, para ser usada por todos los clientes conectados, esto representa un

patrón de diseño “Singleton”. Tenga especial cuidado cuando use la propiedad `LifeCycle` como `Server`, porque la implementación de la clase del lado del servidor necesita ser “thread-safe”, usted debería diseñar esta clase para que pueda ser accedida por múltiples “threads”.

- `Session` es el valor por defecto de la propiedad `LifeCycle`, significa que el Servidor DataSnap crea una clase de servidor por cada cliente conectado.
- `Invocation` en este caso una clase de servidor es creada y destruida cada vez que el cliente invoca un método, por tal motivo el estado de la clase de servidor no se mantiene entre la llamada de los métodos.
- `TDSTCPServerTransport`: este componente implementa el servidor socket “multithread” esperando por las conexiones de los clientes. Este componente no tiene eventos. La propiedad `Port` indica el puerto TCP en el cual él está escuchando, por defecto tiene el valor 211. Es posible usar HTTP(S) para comunicarse entre los servidores y los clientes DataSnap

Principal Componente del lado del Cliente:

- `TSQLConnection`: encapsula una conexión dbExpress a un servidor de Base de Datos. Para el caso de hacer una conexión a un Servidor DataSnap, sólo debemos configurar las siguientes propiedades:
 - `Driver` como `DataSnap` (Desde el punto de vista del cliente es una conexión a una Base de Datos, pero en realidad es una conexión a un servidor DataSnap).
 - `LoginPrompt` como `False` para evitar que aparezca el diálogo de usuario/contraseña cada vez que el cliente se conecte.

Ver Video en el siguiente link: <http://www.youtube.com/watch?v=JFONiqdj0t4>

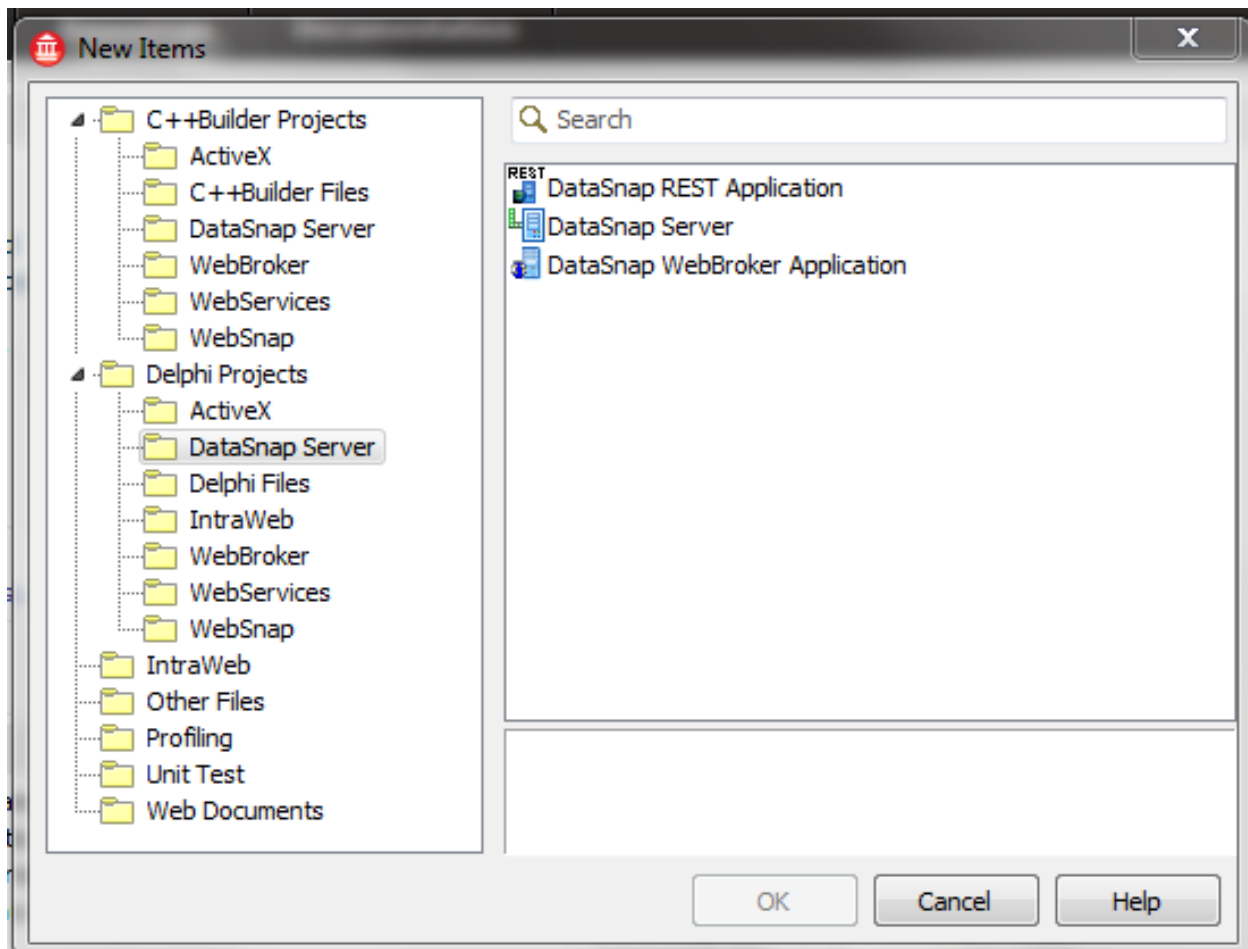
Servidores DataSnap que podemos crear con los Ayudantes

Los asistentes para crear Servidores DataSnap, proveen una manera fácil de implementar una aplicación Servidora usando la tecnología DataSnap. Independientemente del ayudante que seleccione, se crea un ServerContainerUnit que contiene un componente TDSSErver.

Todos los componentes adicionales se enlazan al [TDSSErver](#), usando la propiedad [Server](#) de cada uno de ellos; si el protocolo seleccionado es TCP/IP, la unidad también contiene un componente [TDSTCPSErverTransport](#); si seleccionamos HTTP(S), la unidad incluirá un componente [TDSHTTPSErvice](#).

Ya que nombramos el componente [TDSHTTPSErvice](#), demos la explicación del mismo. [TDSHTTPSErvice](#) permite la invocación de los métodos remotos a través de un cliente usando HTTP(S), utilizando el protocolo REST.

Ver video en el siguiente link: <http://www.youtube.com/watch?v=ALnC67VK8xc>

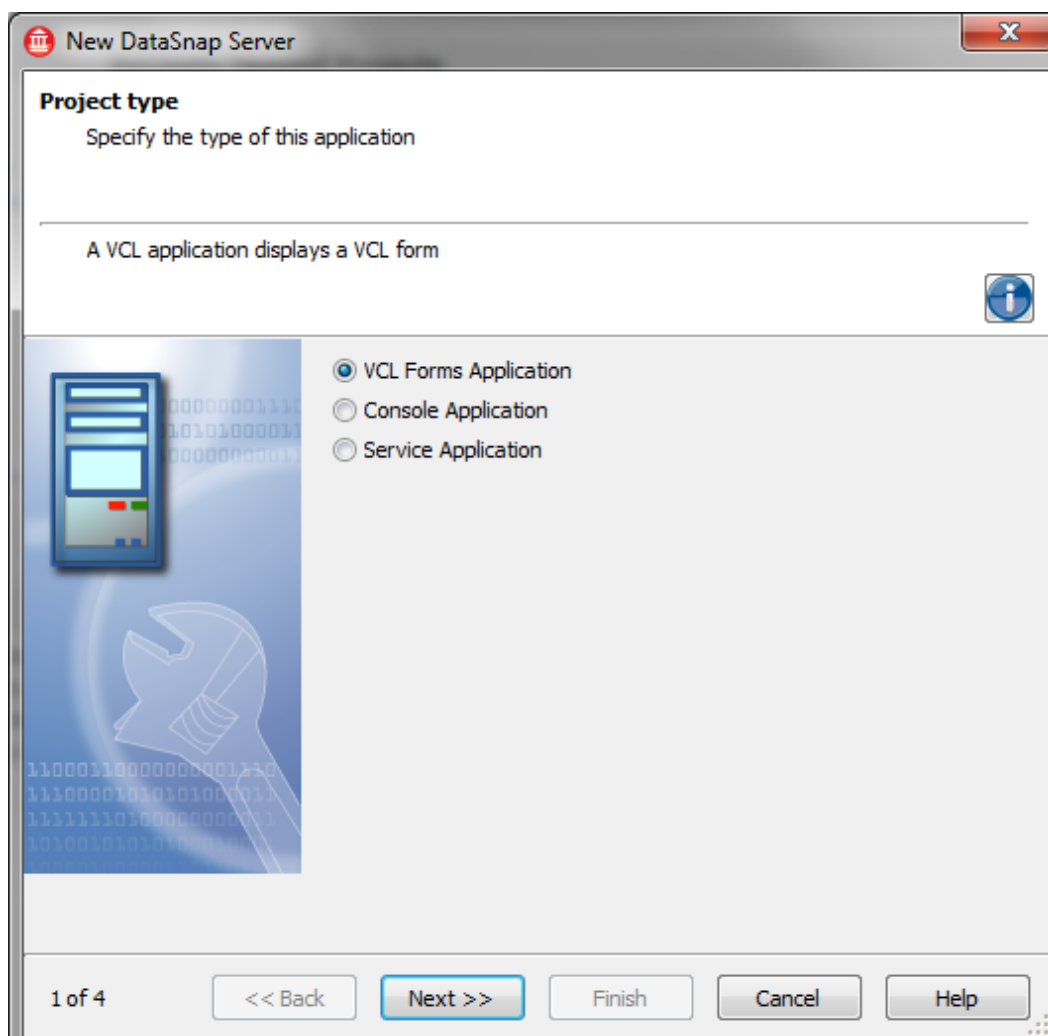


Como notarán en la imagen anterior tenemos tres (03) ayudantes para crear una aplicación Servidora basada en DataSnap:

- DataSnap Server.
- DataSnap REST Application.
- DataSnap WebBroker Application.

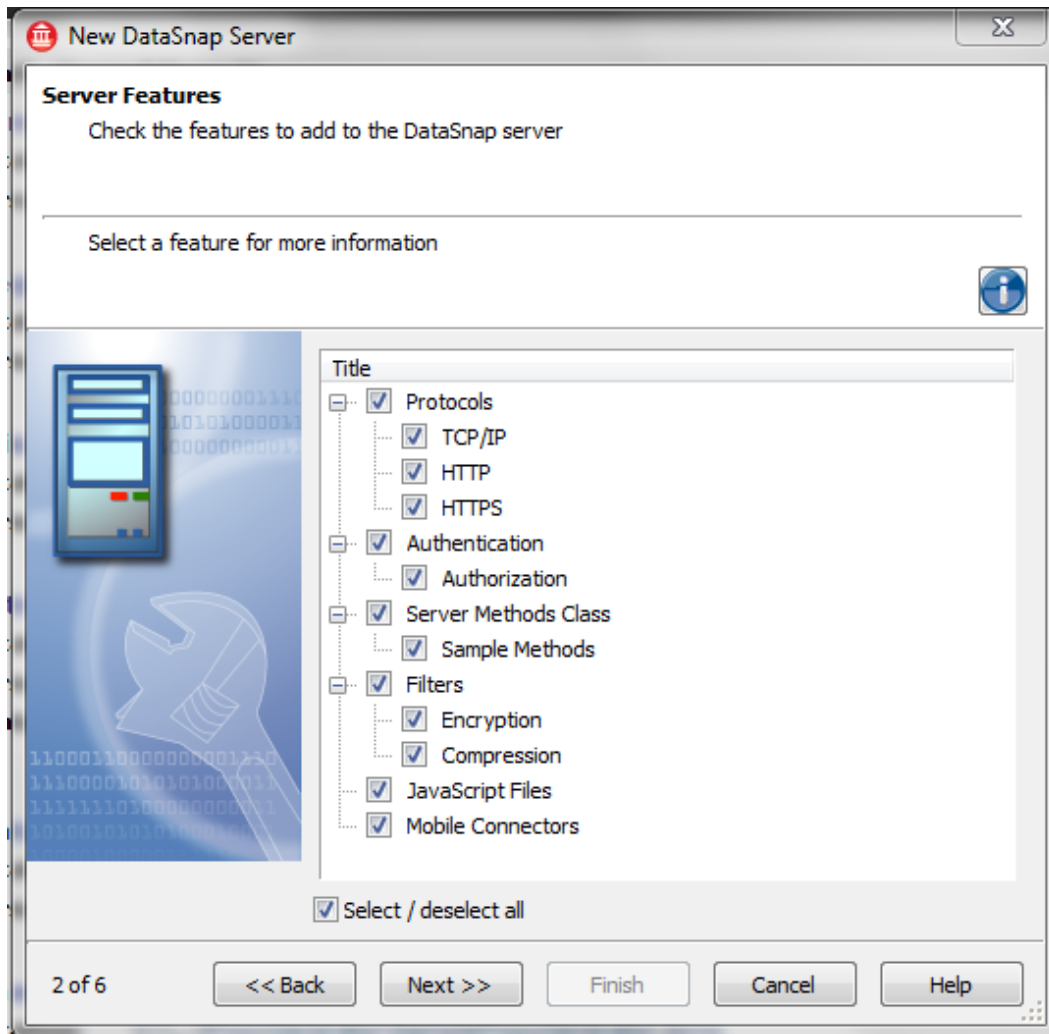
DataSnap Server Wizard:

Tipo de Proyecto:



Podemos seleccionar entre tres tipos de aplicaciones diferentes: **aplicación basada en VCL**, **aplicación de consola** de Windows “cmd” y un **Servicio de Windows**, podemos seleccionar la que más se adapte a nuestras necesidades.

Características del Servidor DataSnap:



En este paso el asistente le permite seleccionar las características que desea agregar al Servidor DataSnap, cada una posee una descripción en la parte superior que aparece cada vez que es seleccionada, que le ayudará a seleccionarla. Los protocolos de comunicación disponibles son: [TCP/IP](#), [HTTP](#) y [HTTPS](#); usted podrá elegir la combinación que desee, si no selecciona ninguna, el protocolo TCP/IP se seleccionará automáticamente. La lista de los puertos disponibles se le mostrará en el paso siguiente.

Si selecciona HTTPS como uno de los protocolos a usar, el asistente le mostrará una página adicional solicitándole información del certificado digital a usar.

Si selecciona la opción "Authentication", se agregará un componente [TDSAuthenticationManager](#) para permitir la implementación de la Autenticación de usuarios.

Al seleccionar la opción "Server Methods Class", el asistente agregará una clase [TDSServerClass](#), clase que permite exponer los métodos del lado del Servidor; si además selecciona la opción "Sample Methods", el asistente agregará dos métodos a la

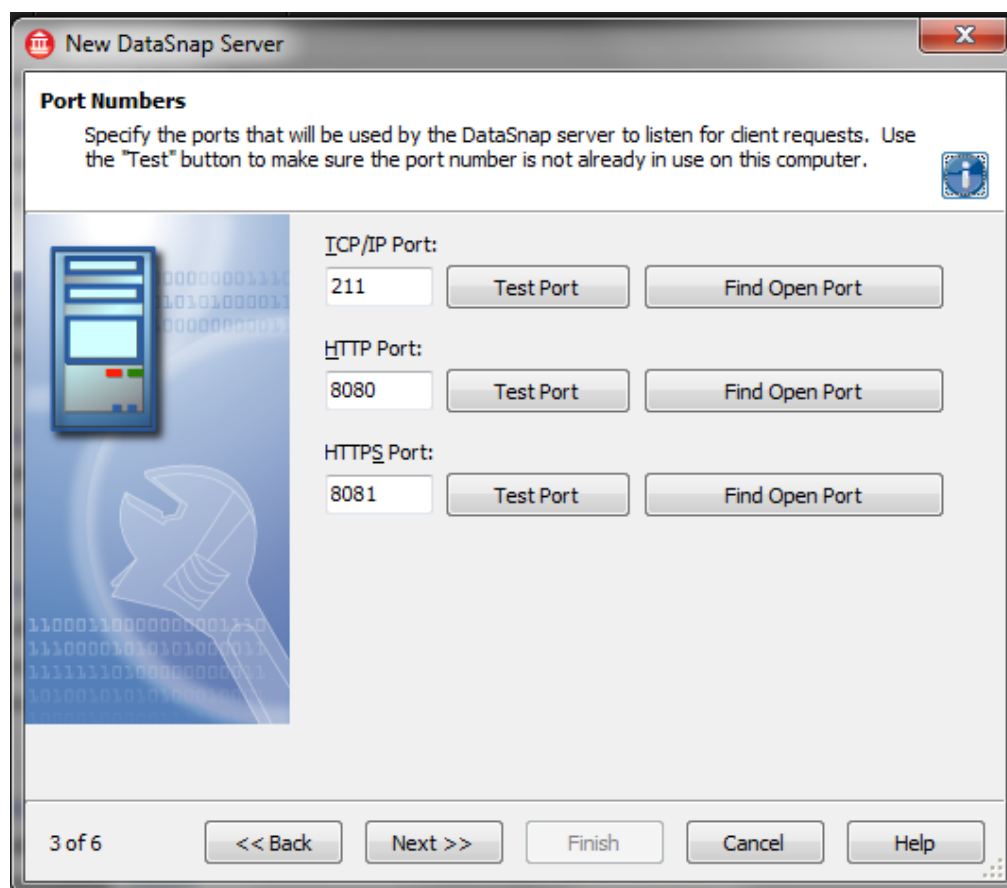
implementación de la clase de Servidor llamados `echoString` y `ReverseString`, los cuales devuelven el valor pasado y el valor invertido respectivamente.

La categoría “Filters” especifica el tipo de filtro que el Servidor DataSnap usará, usted tiene la opción de incluir cifrado y/o compresión.

Los “JavaScript Files” configuran el proyecto con el framework JavaScript y el generador de proxy.

Seleccionando “Mobile Connectors”, le damos soporte a nuestro proyecto para desparchar proxies para dispositivos móviles como Android, Windows Phone y iOS.

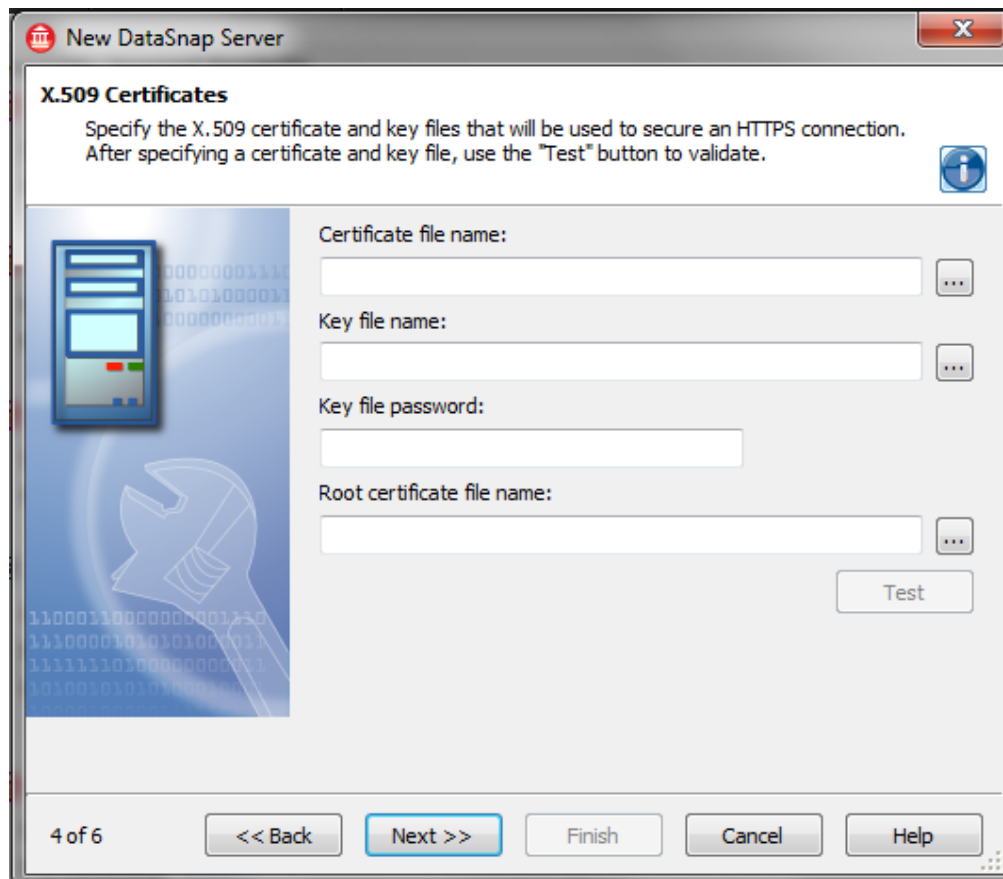
Selección de Puertos:



En este se solicita la selección del puerto TCP/IP de comunicación, adicionalmente el asistente le permite verificar si el puerto se encuentra disponible para ser usado, si desconoce cual o cuales son los puertos disponibles, puede pulsar el botón “[Find Open Port](#)”, el asistente seleccionará un puerto disponible por Usted.

Si selecciona la opción HTTPS, deberá seleccionar también un puerto para la comunicación a través de HTTPS.

Certificados X.509:

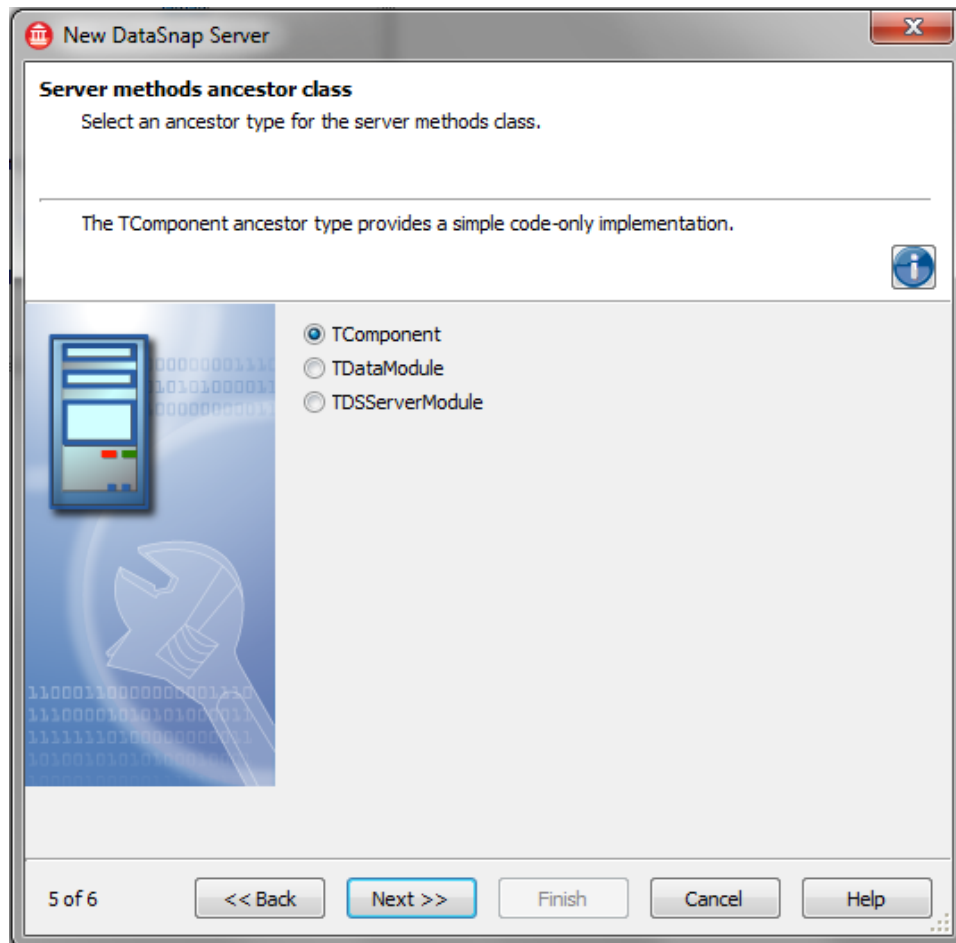


Este paso sólo aparece si se selecciona el uso del protocolo HTTPS, aquí se le solicita que introduzca la información relacionada con el certificado digital X.509:

- Nombre del archivo que contiene el certificado X.509 (*.pem).
- Nombre del archivo que contiene la clave.
- Nombre del archivo que contiene la contraseña del archivo de claves.
- El archivo que contiene el certificado raíz.

Para validar que todo este correcto pulse el Botón "Test".

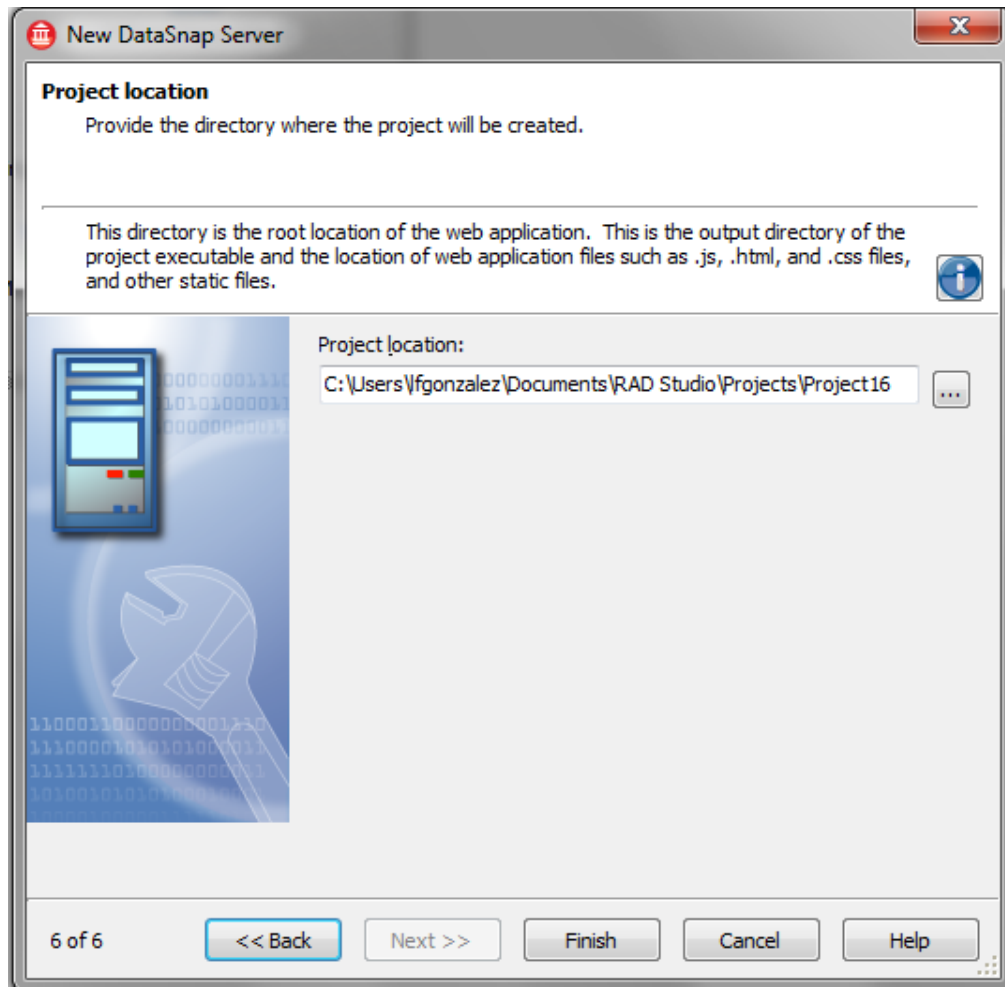
Clase Contenedora para los métodos del Servidor:



En este paso, usted seleccionará que tipo de clase que va a contener los métodos de Servidor.

Seleccione [TDSServerModule](#) si usted desea exponer datasets desde el servidor a sus clientes. Si Usted desea usar componentes no-visuales en la clase del Servidor use [TDataModule](#), si por el contrario Usted desea implementar por completo la clase Servidora debe seleccionar [TComponent](#).

Ubicación del Proyecto:

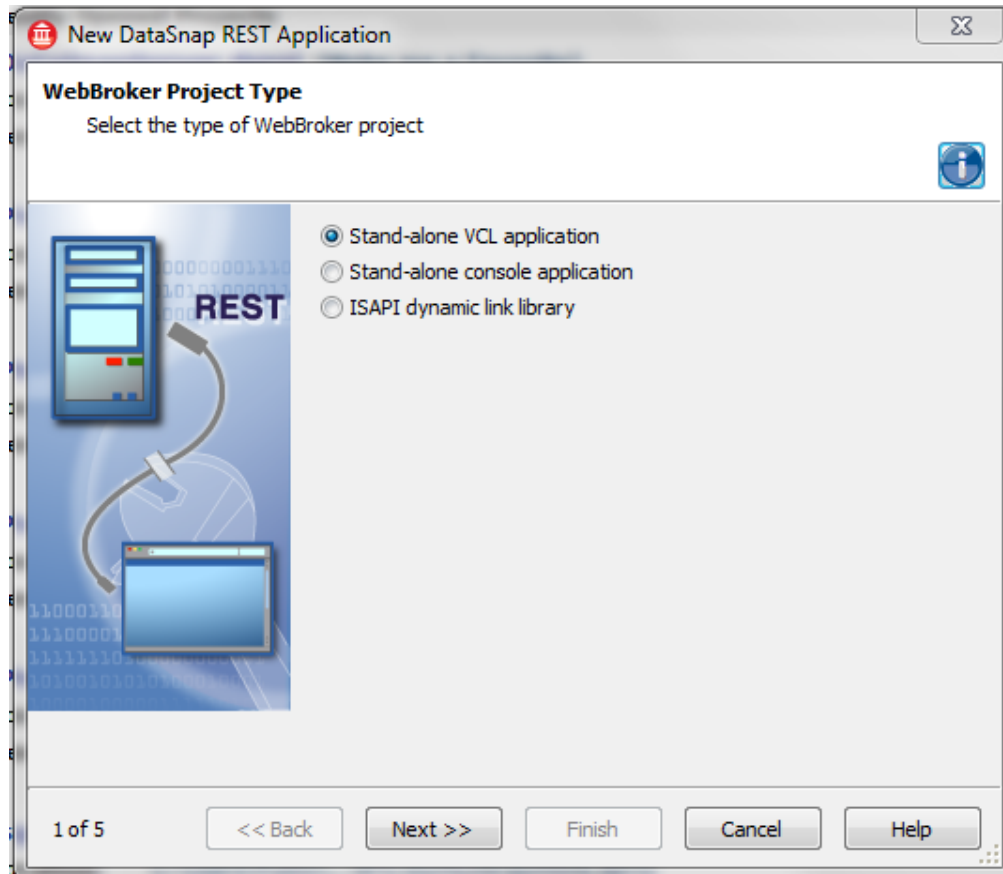


Si Usted seleccionó en el paso 2 (Características del Servidor DataSnap), “JavaScript Files” o “Mobile Connectors”, el asistente le solicitará el directorio raíz de su proyecto web, este directorio tendrá el ejecutable y los archivos web necesarios como: htmls, js, css y archivos estáticos.

DataSnap REST Application Wizard:

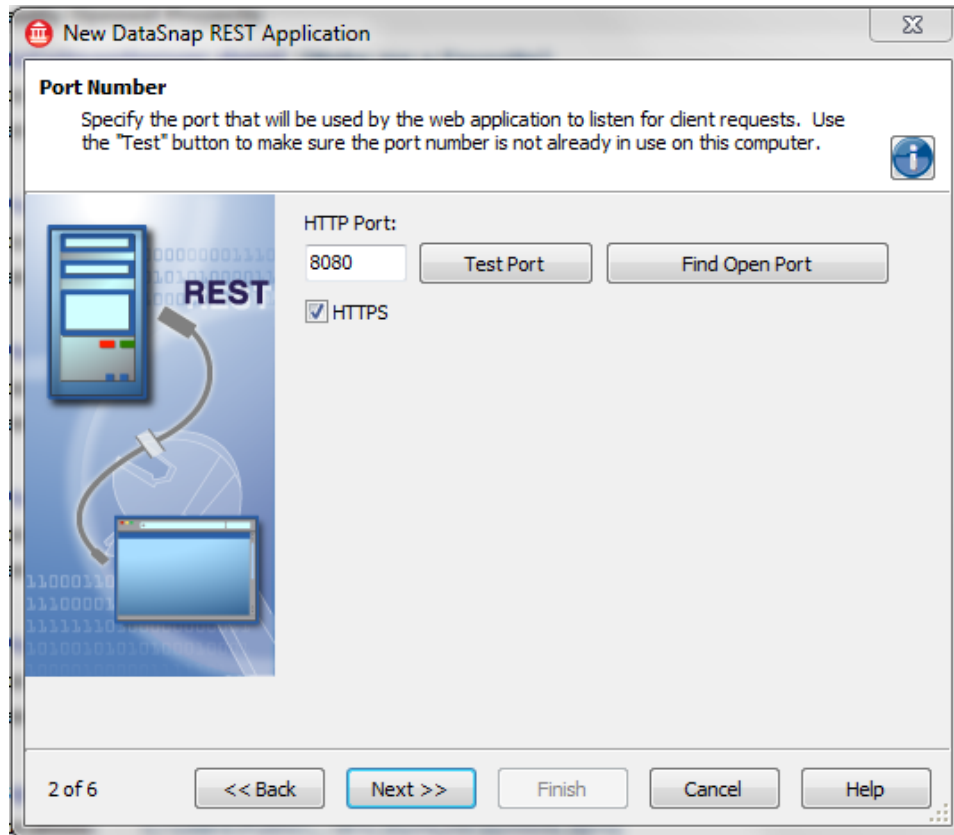
El ayudante DataSnap REST, crea un proyecto que es el punto de partida para construir una aplicación web habilitada para AJAX.

Tipo de Proyecto:



Podemos seleccionar entre tres tipos de Aplicaciones: Una aplicación basada en formularios **VCL**, una aplicación de **consola** y una aplicación **ISAPI**.

Selección de Puerto:



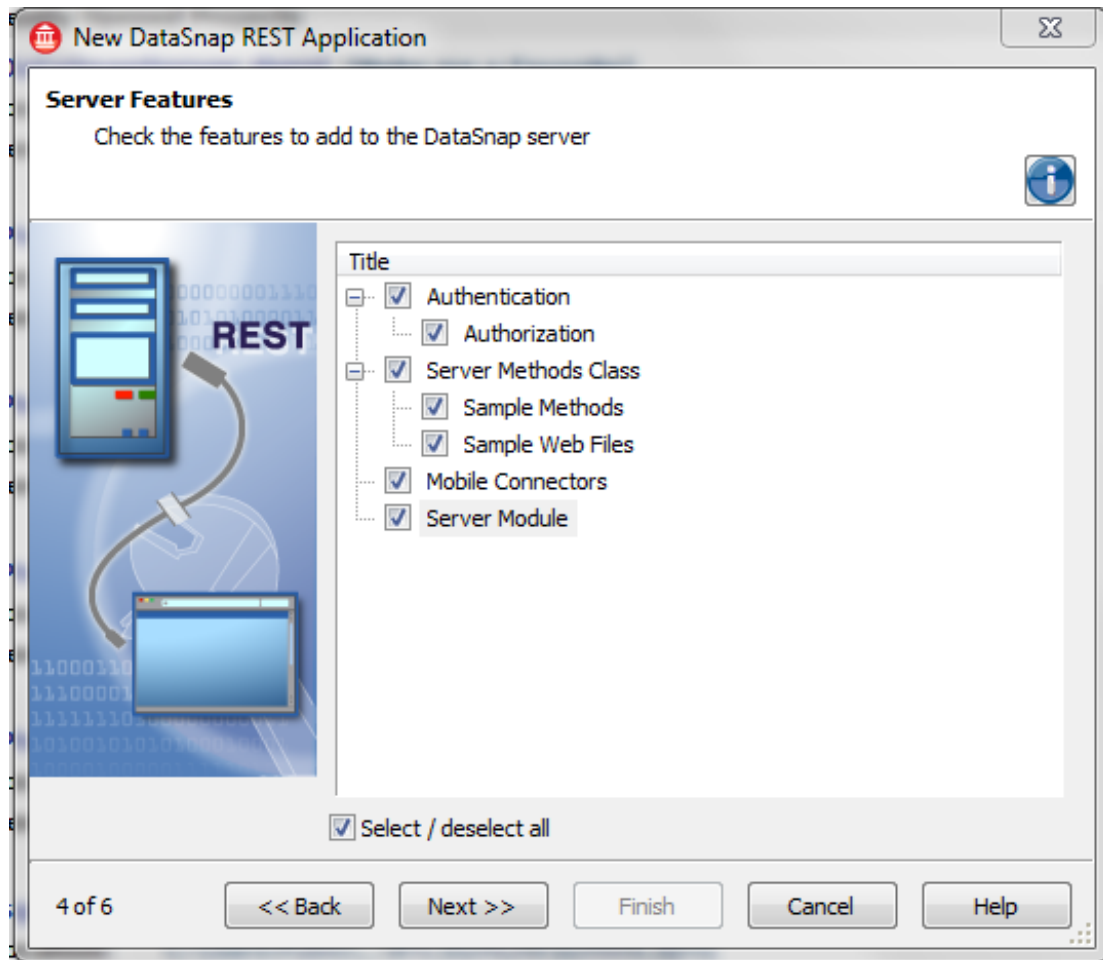
En este paso, seleccionamos el puerto que queremos usar para nuestra aplicación DataSnap REST, esto sólo es posible si en el paso anterior seleccionamos alguna de las opciones: **Stand-alone VCL Application** o **Stand-alone console Application**.

Si seleccionamos **ISAPI**, nuestra aplicación dependerá de un Servidor Web, en nuestro caso **IIS**, lo que significa que este paso no lo veremos.

Certificados X.509:

Ver el paso “**Certificados X.509**”, del Ayudante del **Servidor DataSnap**. Cabe destacar que este paso sólo aparece si el proyecto a crear es diferente de **ISAPI**, si es **ISAPI** los temas relacionados con **HTTPS** dependen del Servidor Web de Windows.

Características del Servidor DataSnap REST:



En este paso usted podrá seleccionar las características que desea para su Servidor DataSnap REST.

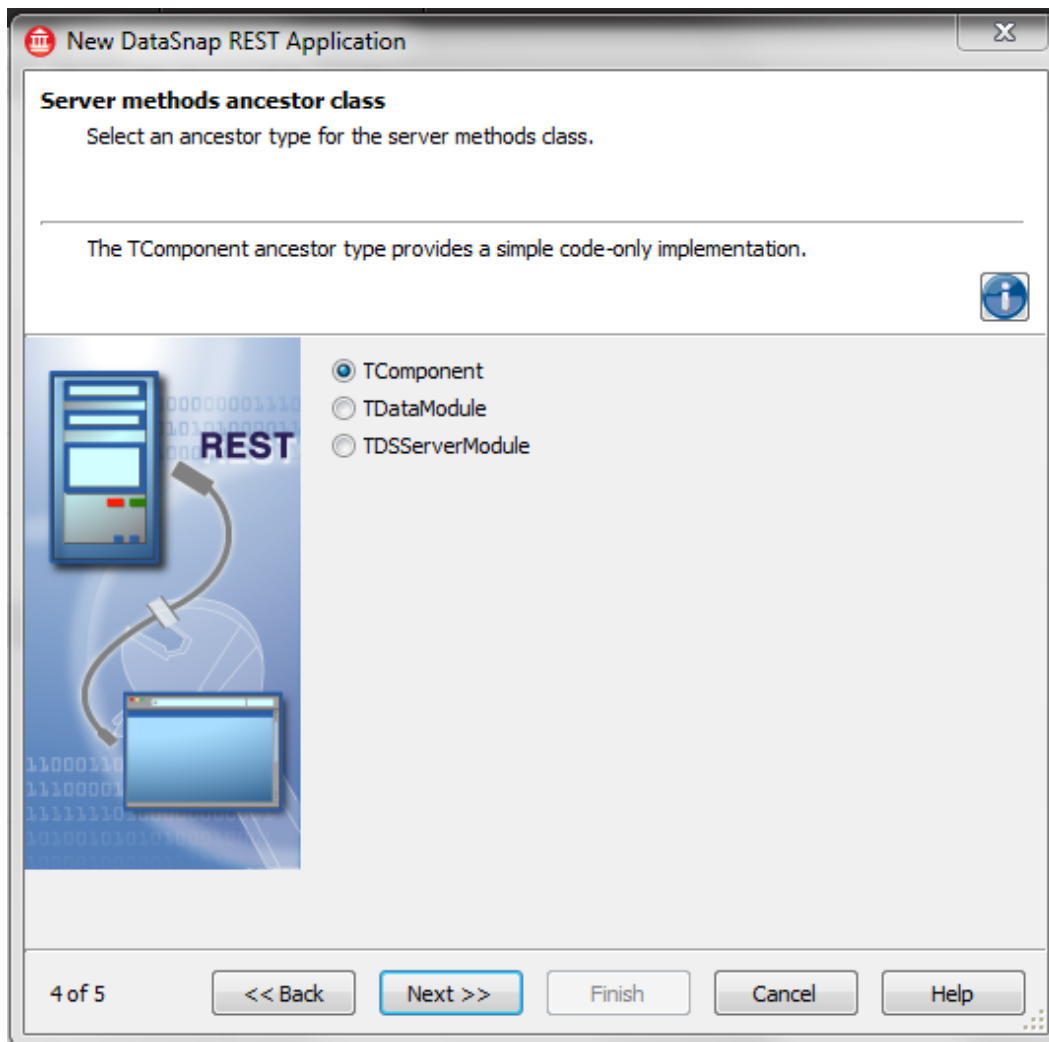
Si selecciona la opción de “[Authentication](#)”, un componente [TDSAuthenticationManager](#) se colocará en el formulario del Servidor, el componente [TDSHTTPWebDispatcher](#) utiliza éste componente para permitir la Autenticación de Usuarios para el Servidor HTTP DataSnap.

Seleccione la opción “[Server Methods Class](#)”, si desea que el asistente cree por usted el componente [TDSServerClass](#) y lo agregue al formulario del Servidor, éste le permitirá definir los métodos que serán invocados remotamente desde los clientes, seleccionando “[Sample Methods](#)” y “[Sample Web Files](#)”, el asistente creará por usted, métodos de ejemplo y páginas web de ejemplos.

La opción “[Mobile Connectors](#)” le permitirá incluir soporte a la generación de proxies para invocar los métodos remotos a través de dispositivos móviles.

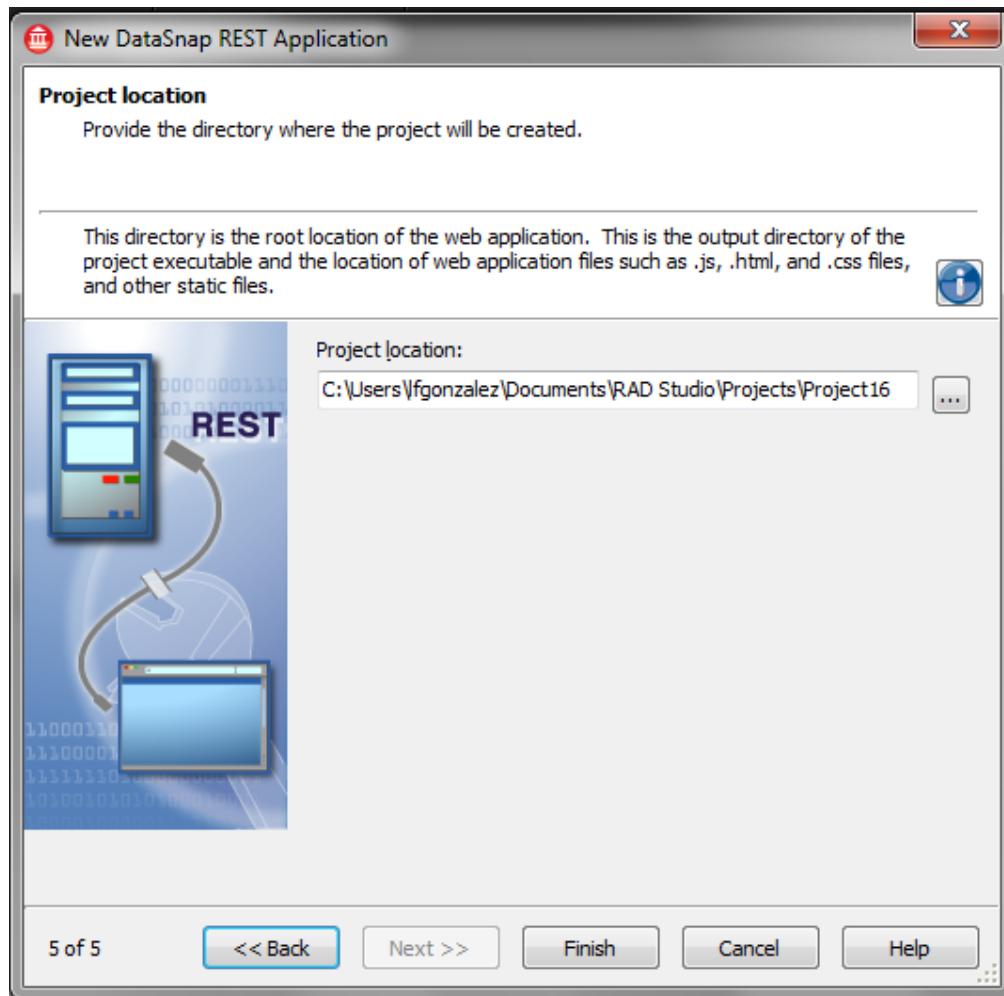
Si selecciona la opción “[Server Module](#)”, usted podrá crear un módulo separado para el Servidor DataSnap; ésto habilitará el soporte a “[heavyweight callbacks](#)” en el Servidor DataSnap REST.

Clase Contenedora para los Métodos del Servidor:



En este paso seleccionamos, la clase padre de nuestra clase contenedora; si deseamos colocar componentes no-visuales seleccionamos [TDataModule](#), si deseamos disponibilizar datasets a los clientes seleccionamos [TDSServerModule](#) y si deseamos crear nuestro métodos de servidor manualmente seleccionamos [TComponent](#).

Ubicación del Proyecto (Archivos Web):

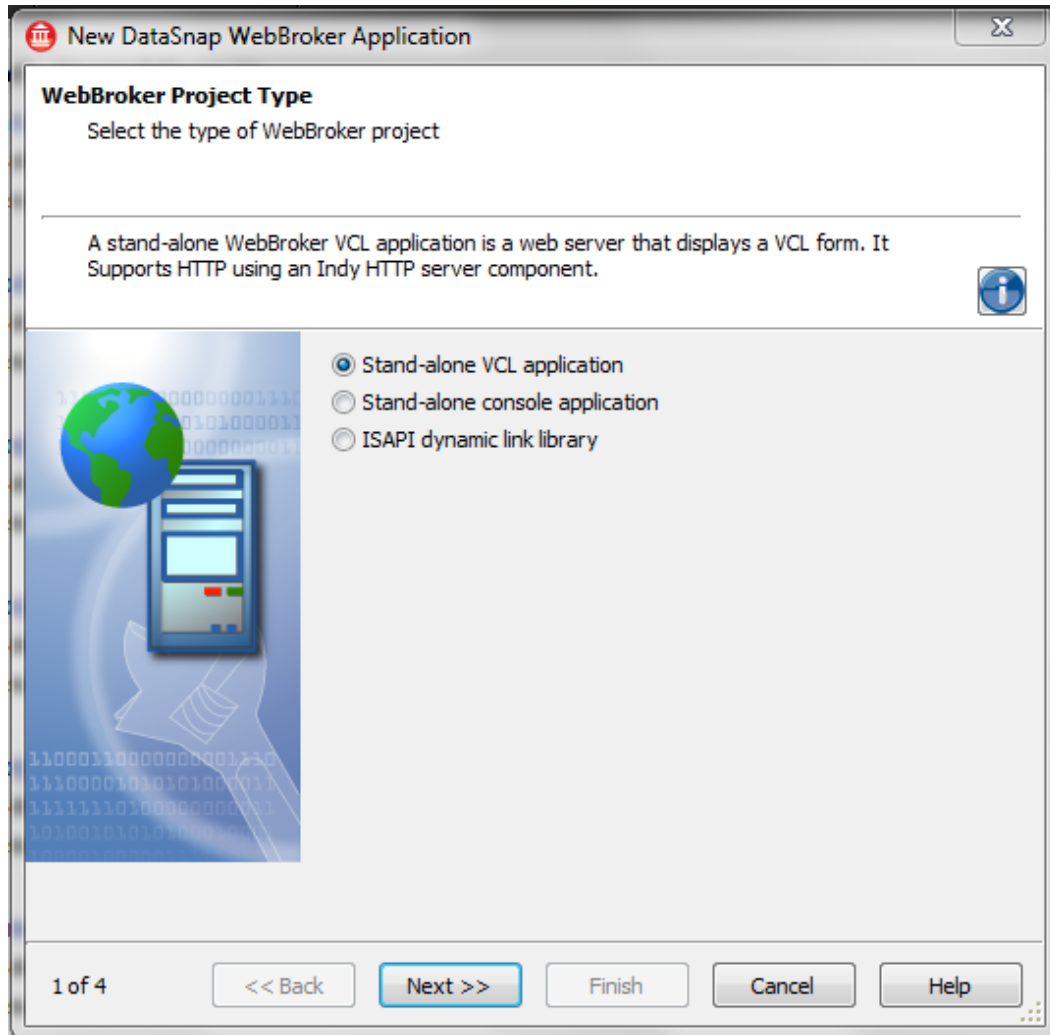


En este paso usted selecciona el root de la aplicación Web, aquí debe estar ubicado el ejecutable y los archivos web disponibles: html, css, images y otros archivos estáticos de tecnología web.

DataSnap WebBroker Application Wizard:

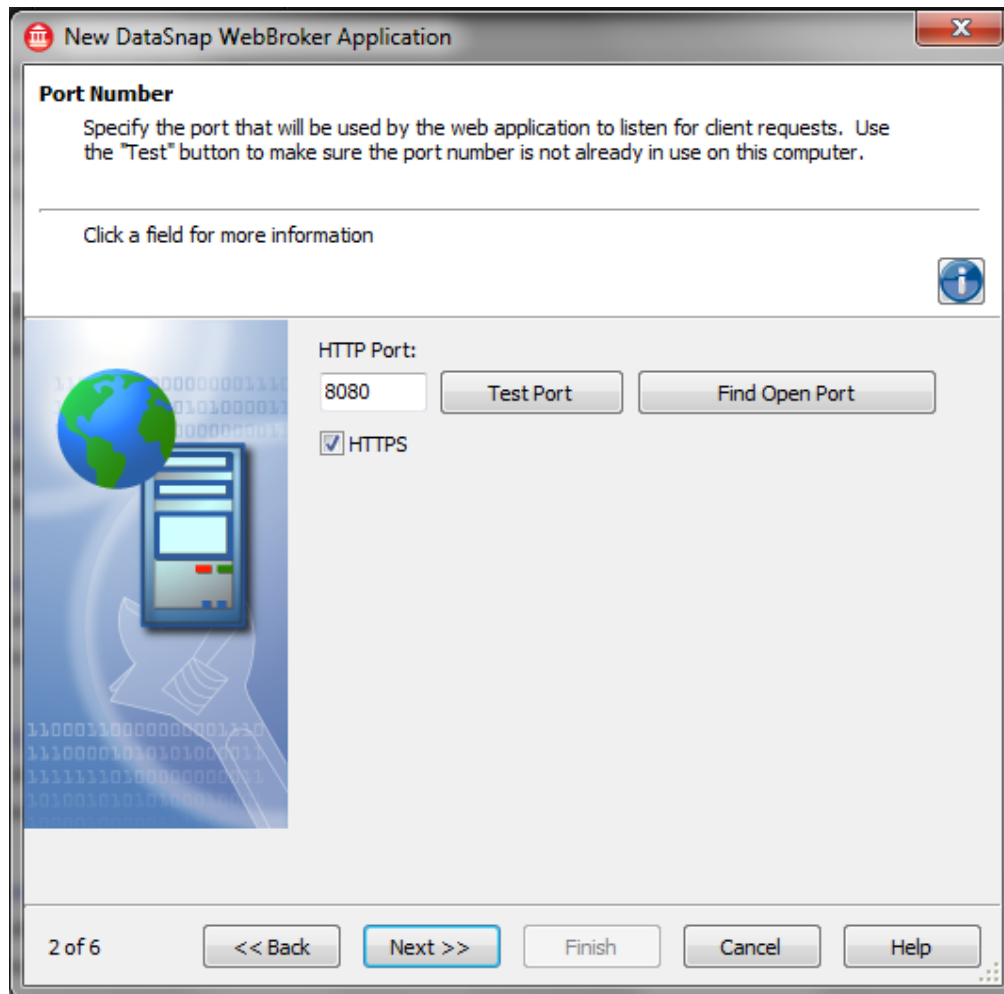
Este asistente proporciona una manera fácil de implementar una aplicación DataSnap utilizando la tecnología WebBroker y la tecnología DataSnap juntas. La Tecnología WebBroker se utiliza para enviar respuestas en formato HTML.

Tipos de Proyecto:



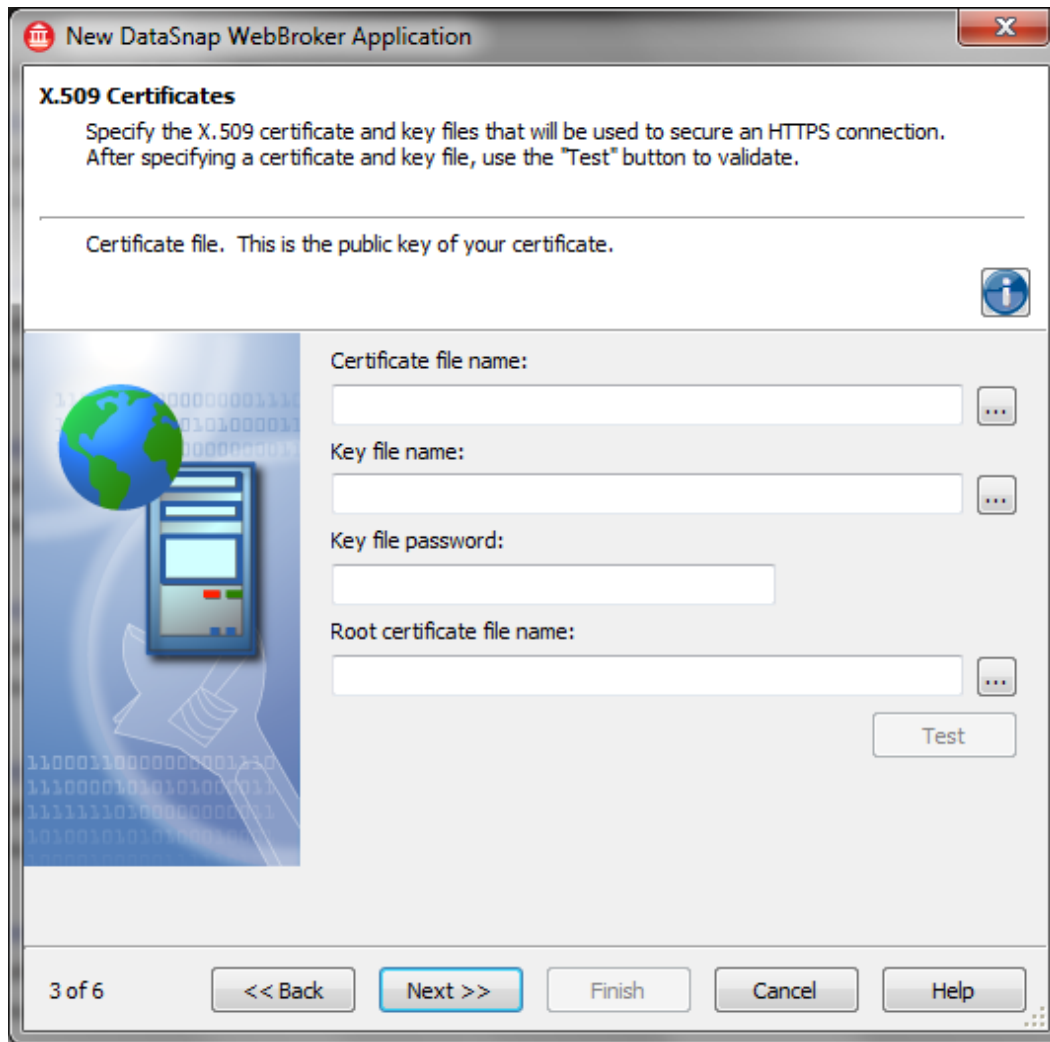
Al igual que el asistente “DataSnap REST”, podemos seleccionar entre tres tipos de proyectos: [Stand-alone VCL application](#), [console Application](#) y [ISAPI dynamic link library](#).

Selección de Puerto:



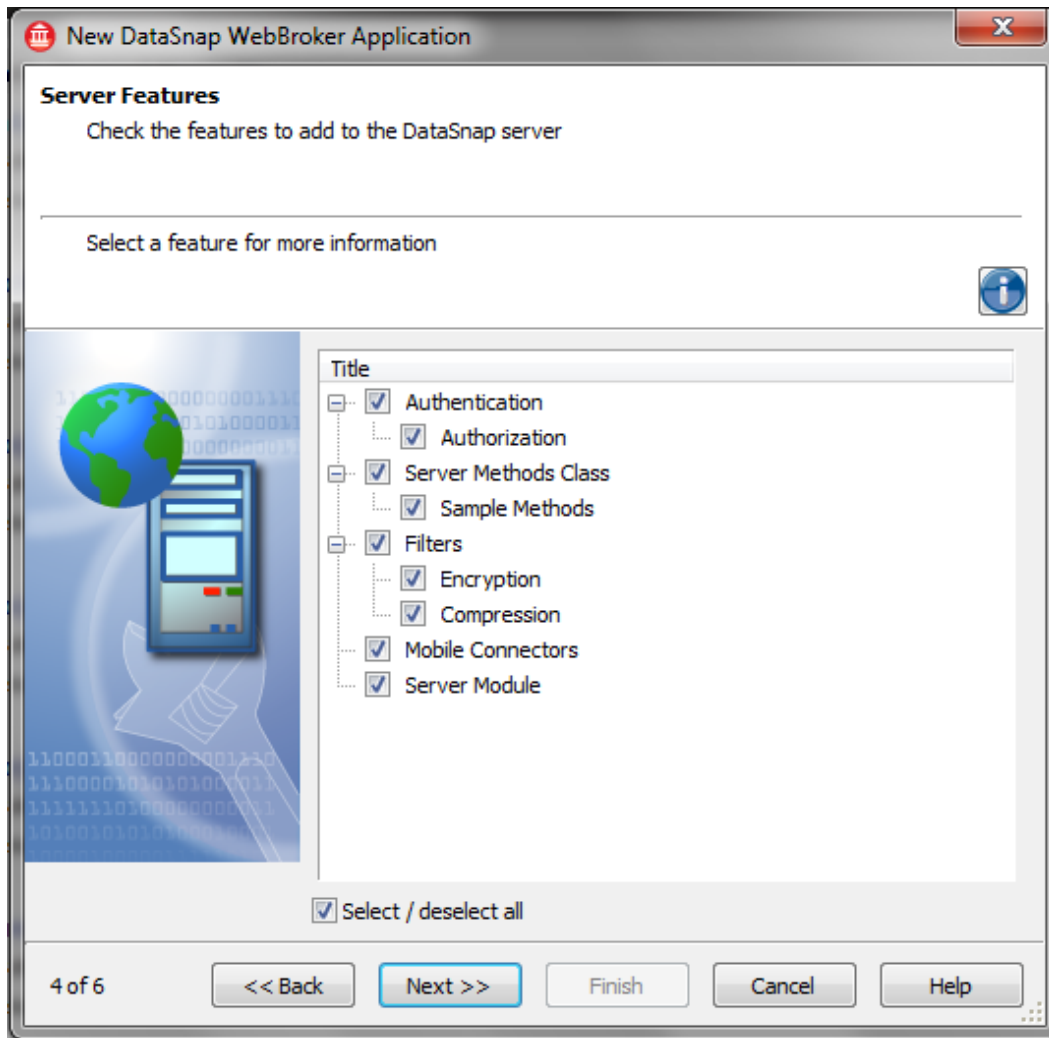
Este paso sólo aparecerá si se selecciona una aplicación [diferente](#) a [ISAPI](#), las aplicaciones ISAPI depende del Web Server por tal motivo, no es necesario seleccionar ningún puerto.

Certificados X.509:



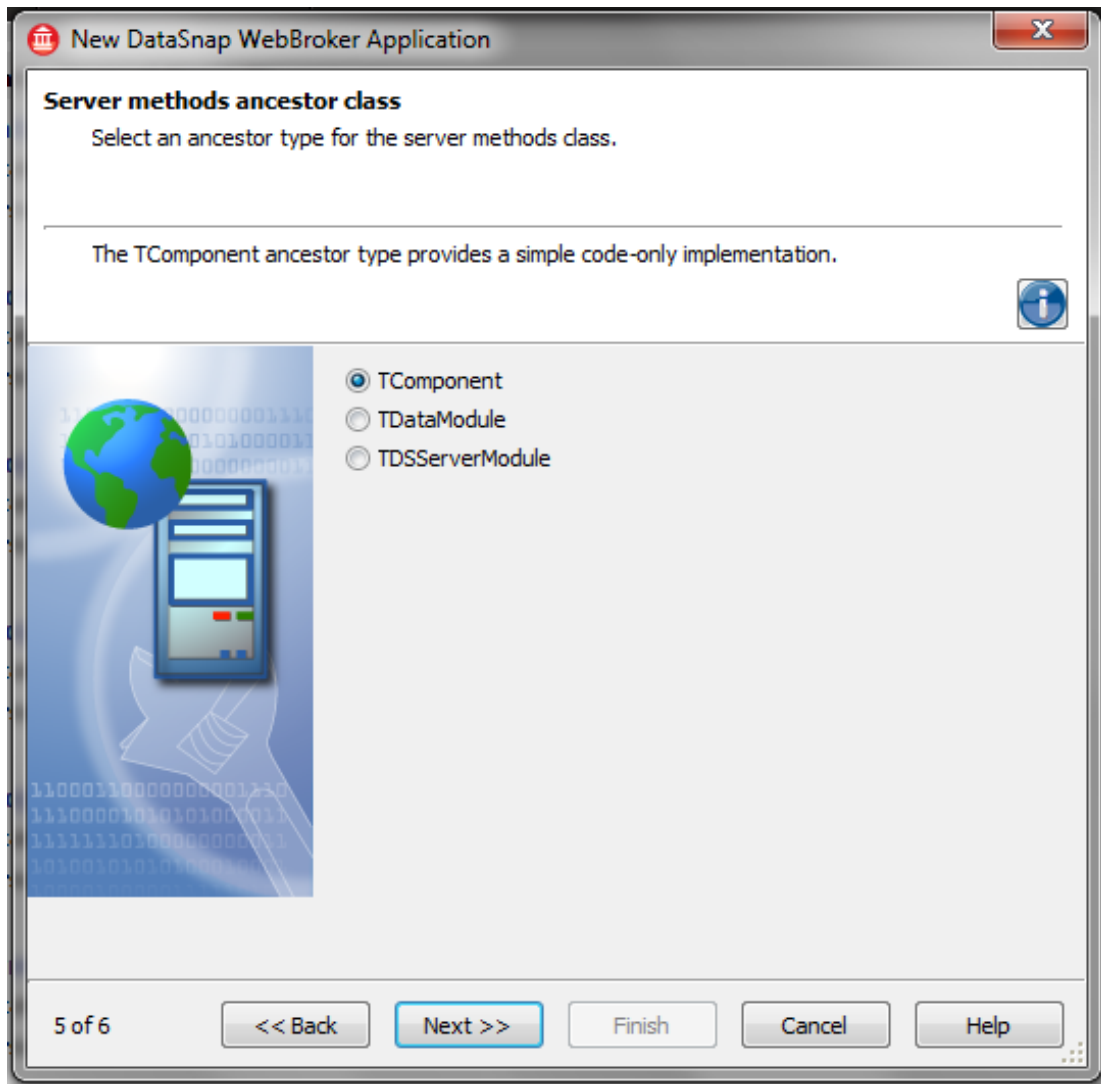
Al igual que el paso anterior sólo aparece si se seleccionado un tipo de aplicación diferente de ISAPI, el tema relacionado con certificados digitales en las aplicaciones ISAPIs es controlado por el Servidor Web, en el caso de Windows IIS.

Características del Servidor:



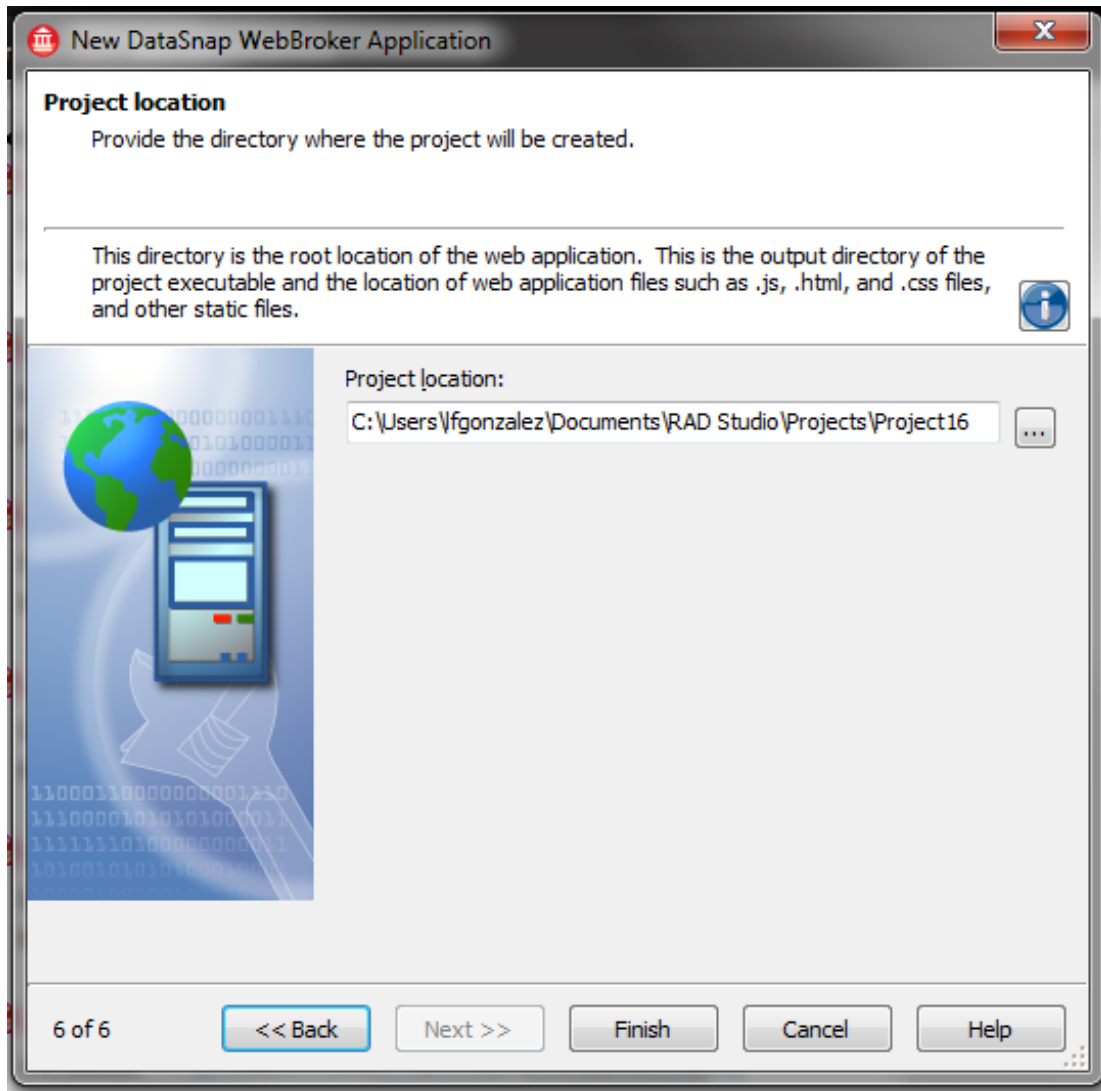
En este paso seleccionamos la características que deseamos que tenga nuestro Servidor DataSnap. Ver explicación en los Asistentes Anteriores.

Clase Contenedora para los Métodos del Servidor:



Aquí podemos seleccionar el padre de la clase que contendrá los métodos del Servidor. Si deseamos usar componentes no-visuales seleccionamos [TDataModule](#), si deseamos exponer datasets para nuestros clientes usamos [TDSServerModule](#) y si queremos simplemente hacer métodos remotos seleccionamos [TComponent](#).

Ubicación del Proyecto (Archivos Web):





Este directorio es el raíz de nuestra aplicación web.

Componentes de la paleta “DataSnap Server”

Imagen	Nombre	Descripción
	TDSProxyGenerator	Componente que implementa el Generador de Proxy del DataSnap Server
	TDSServerMetaDataProvider	Implementa un proveedor de metadatos que se comunica con el proceso del TDS Server.
	TDSConnectionMetaDataProvider	Implementa la conexión que provee los metadatos para el Servidor DataSnap
	TDSRestConnection	Provee la conexión para un Servidor DataSnap REST
	TDSRestMetaDataProvider	Implementa un proveedor de metadatos de un proveedor DataSnap REST
	TDS Server	Este componente gestiona la creación y el ciclo de vida de los “Sockets” y las clases de Servidor
	TDS Server Class	Componente que especifica la clase servidora que contiene los métodos publicados, métodos que se pueden llamar remotamente desde los clientes utilizando la invocación de métodos dinámicos.
	TDS HTTP Service	Permite la invocación de los métodos del Servidor utilizando el Protocolo REST
	TDS Cert Files	Proporciona información acerca de los certificados X.509 y sus claves privadas
	TDS Authentication Manager	Permite implementar la Autenticación (Usuario/Contraseña) en un Servidor DataSnap
	TDS Client Callback Channel Manager	Se encarga de los “callbacks” que se encuentran registrados en los Servidores DataSnap
	TDS HTTP Service File Dispatcher	Es el responsable de responder las peticiones HTTP de Archivos
	TDS HTTP Service Web Dispatcher	Ofrece servicios HTTP ligero para la tecnología DataSnap e implementa protocolos de Internet como REST
	TDS HTTP Service Proxy Dispatcher	Ofrece generación de proxy para los Servidores DataSnap que no están basados en WebBroker. Se utiliza con “DataSnap Mobile Connectors” para manejar las peticiones HTTP que requieren de proxies específicos.

Embarcadero Delphi XE 3. Desarrollando Aplicaciones con DataSnap

Imagen	Nombre	Descripción
	TDSProxyDispatcher	Es el generador de proxy para Servidor basados en WebBroker.
	TDSTCPServerTransport	Es un transportador basado en Socket para el TDSServer

Ejemplos de Servidores y Clientes DataSnap usando los Asistentes

Para estos ejemplos aprovecharemos los asistentes y veremos un ejemplo de cada uno de los posibles Servidores basados en DataSnap. Hablaremos de los componentes que los asistentes colocan automáticamente dependiendo de la selección de las características del Servidor.

1. DataSnap Server. Ver video <http://www.youtube.com/watch?v=qnvWIgc1SQA>