PFC .NET – Ingeniería Informática – UOC

Proyecto: Flowerpot

Documento: PEC 3. Implementación.

Autor: David García Casado (dgarciacas@uoc.edu)

Tutor: Juan Carlos González Martín

Fecha: 23/05/2011

ÍNDICE

Introducción

El presente documento se adjunta a la 3ª entrega del PFC del área .NET desarrollado conforme a la propuesta con título "Iluminando el escritorio, la web y la nube con Microsoft Silverlight" y recoge los comentarios relativos a la implementación.

Comentarios de la implementación

La aplicación está basada en el template de Visual Studio 2010 llamado "Silverlight Business Application". La solución se divide en un proyecto Web (flowerpot.Web) y en la aplicación de Silverlight propiamente dicha (flowerpot).

La base de datos está creada en Microsoft SQLServer 2008 R2. La implementación, en mi caso, ha sido sobre una instancia de SQLEXPRESS. Para ello he usado un script de creación de tablas.

Para crear el modelo de entidades he usado Entity Framework de .NET, y he generado el modelo a partir de la base de datos .

Para el modelo de negocio he usado una Domain Service Class, que automaticamente, genera los métodos esenciales para el acceso a las entidades. En este ámbito, he hecho varias modificaciones para como parte de la lógica de mi aplicación.

En cuanto a recuperar entidades en función de parámetros he añadido en la clase flowerpotDomainService.cs:

- GetActualizacionEjemplar(int idEjemplar): Actualizaciones de un ejemplar.
- GetComentarioEjemplar(int idEjemplar): Comentarios de un ejemplar.
- GetEjemplarEspecie(int idEspecie): Ejemplares de una especie.
- GetEjemplarUsuario(int idUsuario): Ejemplares de un usuario.
- GetLogin(string Nombre, string Password): Verifica el login al sistema.
- GetUsuarioId(int idUsuario): Datos de un usuario en concreto.

En cuanto a validación de datos he configurado en la clase flowerpotDomainService.metadata.cs las siguientes restricciones, incluyendo el texto que debe mostrar la aplicación en caso de que no se cumplan:

Propiedad Observaciones requerida en la entidad Actualizaciones:

```
[Required(AllowEmptyStrings = false, ErrorMessage = "Debe rellenar el campo Observaciones")]
```

Propiedad Texto requerida en la entidad Comentarios:

```
[Required(AllowEmptyStrings = false, ErrorMessage = "Debe introducir un Comentario")]
```

Propiedad Etiqueta requerida en la entidad Especies:

```
[Required(AllowEmptyStrings=false,ErrorMessage="Debe rellenar el campo Etiqueta")]
```

Propiedad email bien formado en la entidad Usuarios:

```
[RegularExpression(@"^([\w-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$",
ErrorMessageResourceName = "ValidationErrorInvalidEmail",
ErrorMessageResourceType = typeof(ValidationErrorResources))]
```

Propiedad Name no repetido de la entidad Usuarios:

```
[CustomValidation(typeof(flowerpot.Web.Services.UserValidator), "AsyncValidateUser")]
```

Para esta última he usado una validación personalizada implementada en la clase UserValidator, que hace uso de la función boolean IsUserExist(string userName) definida en flowerpotDomainService.

En la aplicación Silverlight, las principales cuestiones a tener en cuenta son el uso de la clase Autenticado para almacenar si el usuario está validado en el sistema, si es propietario de los ejemplares a los que accede y si es un administrador.

En base a esta información, se van a habilitar/deshabilitar los botones de modificación/borrado de Especies y Ejemplares (incluyendo los comentarios y actualizaciones) y se va a restringir la gestión de la información de los usuarios.

Las permisos se resumen en la siguiente tabla:

	No Login	Login	Administrador
Especies Consultar	X	X	X
Especies Añadir/Editar/Borrar		X	X
Ejemplares Consultar	X	X	X
Ejemplares Añadir		X	X
Ejemplares Editar/Borrar		Propietario	X
Actualizaciones Consultar	X	X	X
Actualizaciones Añadir/Editar/Borrar	X	Propietario	X
Comentarios Consultar	X	X	X
Comentarios Añadir		X	X
Comentarios Editar/Borrar		Propietario	X
Usuarios Alta	X	X	X
Usuarios Consultar/Editar/Borrar		Propietario	X

El Workflow de la aplicación consiste en que en la pantalla de inicio se selecciona un ejemplar en base a una especie o a un usuario y desde allí se pasa a la pantalla de Comentarios o Actualizaciones con el ID del ejemplar como parámetro en la url. Si esto lo hace un usario autenticado, guardo en la clase Autenticado si dicho usuario es el propietario del ejemplar, puesto que hay que tenerlo en cuenta a la hora de mostrar los botones como he indicado en el párrafo anterior.

La clase Autenticado se actualiza cuando se hace Login o Logout.

La tabla con todos los usuarios sólo es mostrada a los administradores. Un usuario logueado sólo podrá ver/modificar o borrar sus propios datos. La propiedad Administrador de la ficha de usuario sólo es modificable por los administradores.

La consulta de la información está basada en grids y los mantenimientos en pantallas ChildWindow que cargo a partir de la entidad seleccionada en cada momento en el grid.

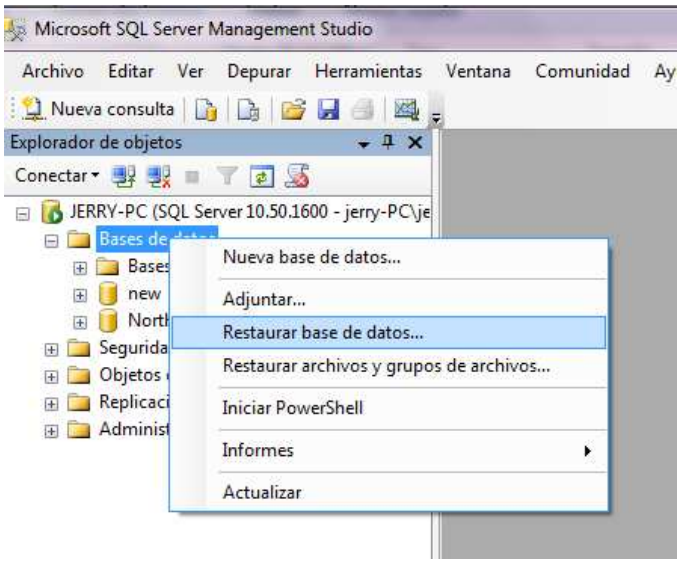
La única salvedad se hace en el mantenimiento de los ejemplares, en el que antes de abrir la ventana para Editar, cargo las especies y los usuarios para que se muestren en un ComboBox. Si se hace una edición, hay que tener en cuenta que hay que mostrar el ComboBox con la especie y el usuario correspondiente ya seleccionados. Y si se hace un alta, el Identificador del Usuario será el del usuario logueado en la aplicación en ese momento.

Configuración para su uso

Base de datos SQLServer

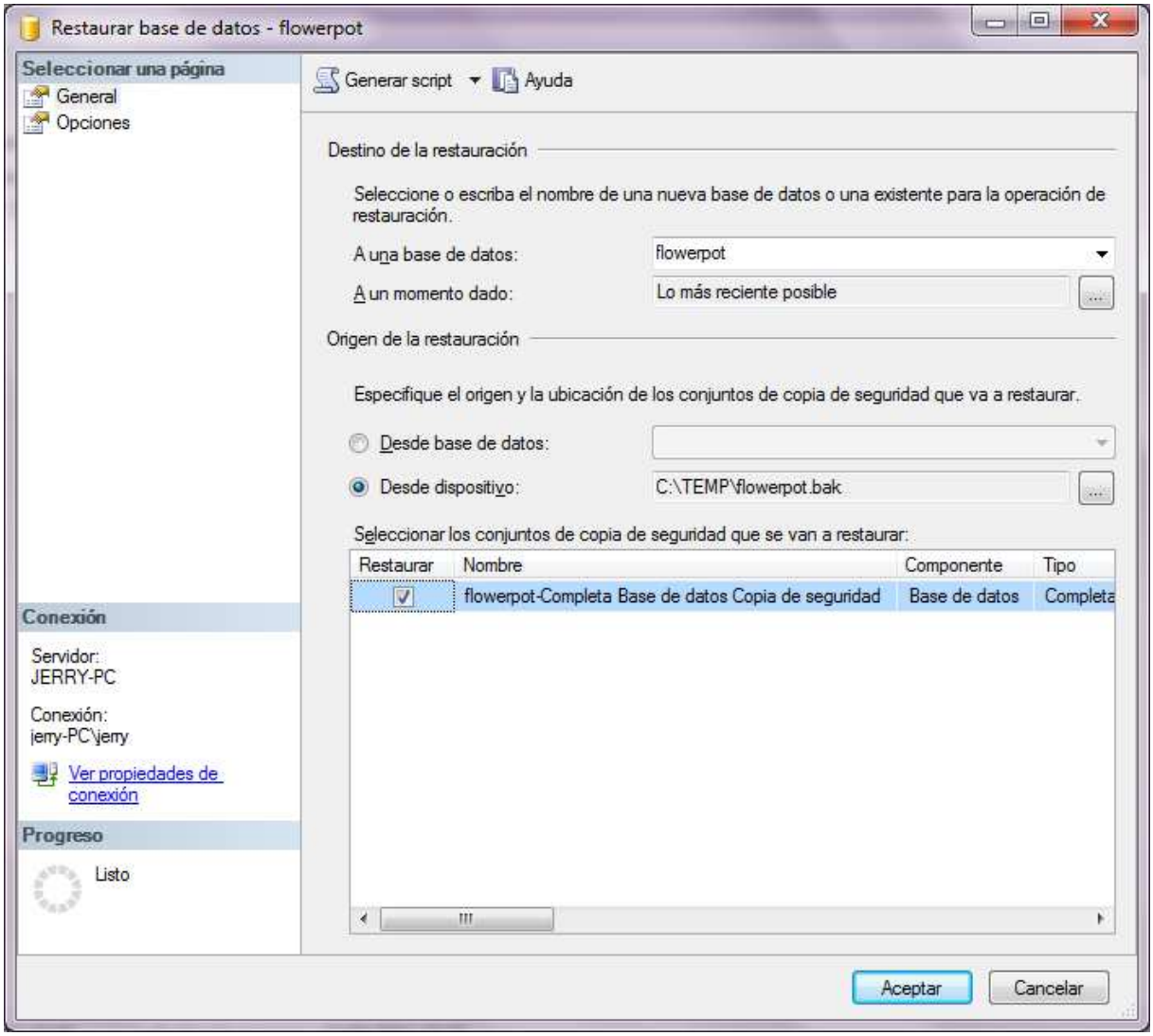
Copiar el fichero flowerpot.bak al directorio c:\temp.

Botón dcho. Sobre Bases de datos → Opción Restaurar base de datos:



Seleccionar las opciones que se ven en la siguiente imagen:

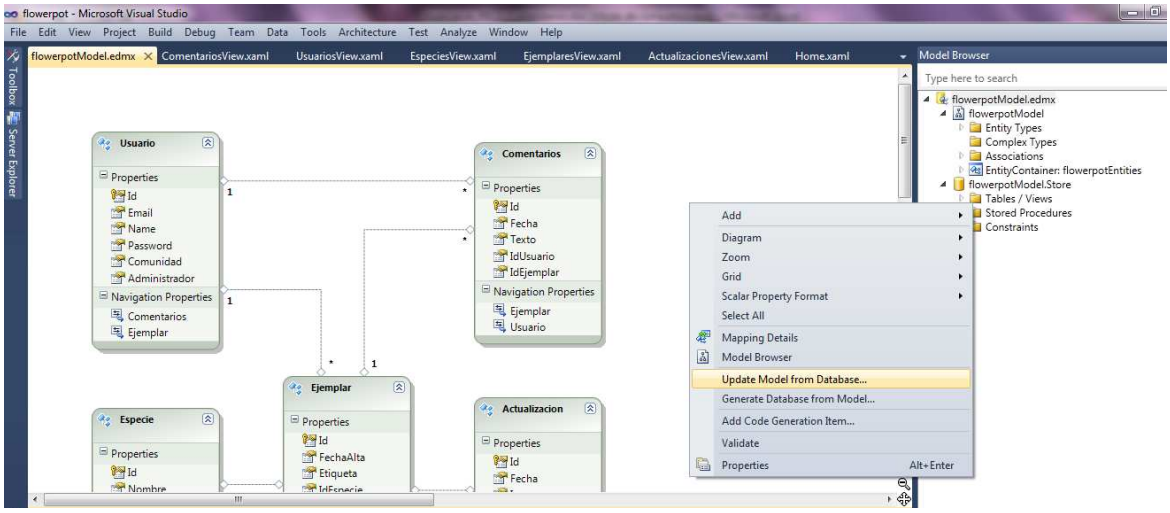
1. A una base de datos con nombre flowerpot.
2. Desde dispositivo (seleccionar c:\temp\flowerpot.bak).
3. Marcar Restaurar.
4. Pulsar Aceptar.



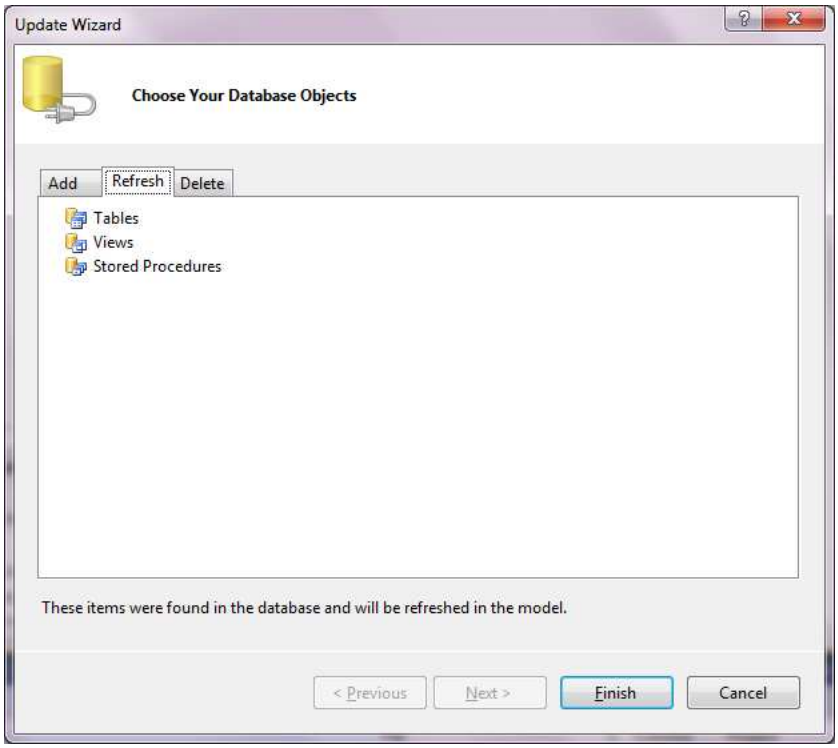
Visual Studio

En el proyecto flowerpot.Web de Visual Studio, hacer doble click sobre flowerpotModel.edmx.

Pinchar con el botón dcho. Sobre el área de diseño de la base de datos y seleccionar la opción "Update Model from Database" (Ver imagen):



En el asistente, seleccionar la pestaña Refresh y pulsar Finish.



Acceso a la aplicación

Existen ciertas funcionalidades que no requieren autenticación y que se pueden usar al arrancarla. Como he explicado anteriormente, la mecánica es seleccionar una especie o un usuario de alguno de los Combos, y pinchar en uno de los ejemplares del DataGrid.

Pulsando en Actualizaciones se consiguen visualizar las Actualizaciones que un usuario hace de sus ejemplares y pulsando en Comentarios, se visualizan los Comentarios que cualquier usuario hace de cualquier ejemplar. Una vez logueado, el usuario podrá añadir/editar/borrar actualizaciones y comentarios.

Para registrarse como nuevo usuario hay que pinchar en el enlace Usuarios de la parte superior y después en el botón Registrar.

El administrador accede con el usuario admin y la contraseña admin. Un usuario autenticado en la aplicación es luis con la contraseña luis.

Para acceder como usuario hay que pinchar en la palabra login de la parte superior.

Para crear/modificar/borrar una Especie se selecciona el enlace Especies de la parte superior. De forma análoga para crear un ejemplar.