Artículo técnico: AX2012 – Herencia de tablas (II)

Revisión: 0.1
Fecha: 08/03/2012
Autor: Manel Querol (mquerol@gmail.com)

http://www.trucosax.com

Introducción

Esta es la segunda y última parte del artículo en el que intentamos analizar y explicar la nueva característica técnica de AX2012 que nos permite usar la herencia de tablas.

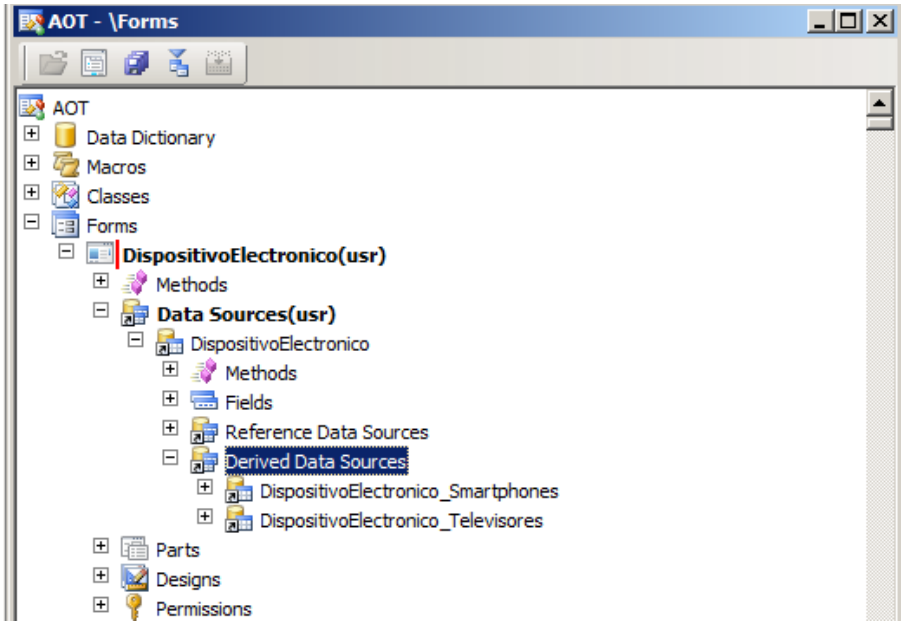
En el primer artículo explicamos un poco la parte teórica y los entresijos de cómo actúa el sistema por dentro y en esta segunda entrega nos centramos en explicar como se usa la herencia de tablas en nuestros formularios, procesos, etc.

¿Cómo se usa?

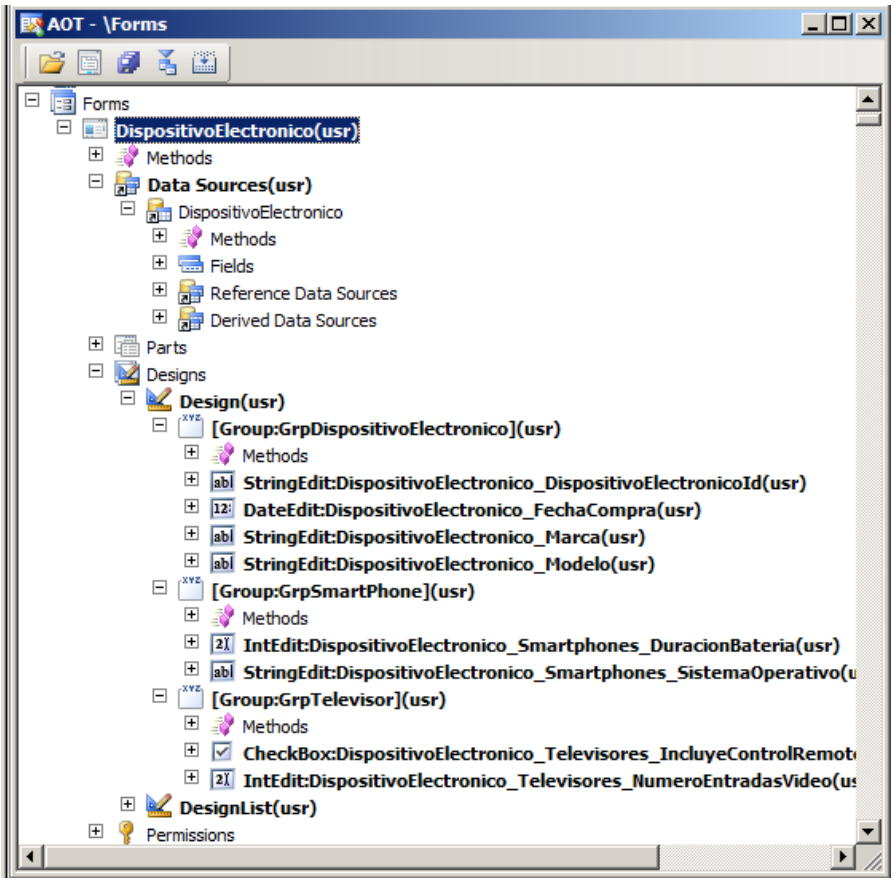
Ahora que ya hemos visto un poco por encima como funciona internamente el sistema de herencia de tablas, vamos a ver con un pequeño ejemplo como puede utilizarse la herencia de tablas.

Creamos un formulario llamado "DispositivoElectronico" y arrastramos en su zona de "DataSources" nuestra tabla de Dispositivos electrónicos.

Podemos observar, como se muestra en la imagen, que nuestra tabla de Dispositivos electrónicos no ha venido sola a este formulario sino que se ha traído consigo sus dos tablas derivadas.



Ahora creamos tres grupos en el diseño (un grupo por cada una de nuestras tres tablas) e insertamos en cada uno de ellos los controles de edición de los campos pertenecientes a cada una de las tablas respectivamente (arrastramos los campos de los "datasources" al interior de los grupos) .



Establecemos a TRUE la propiedad “AutoDeclaration” de los grupos GrpSmartPhone y GrpTelevisor.

Cuando ejecutamos el formulario vemos que nos muestra los datos comunes a todos los dispositivos y los datos concretos de cada una de las tablas derivadas según sea el caso en los respectivos grupos que hemos creado.

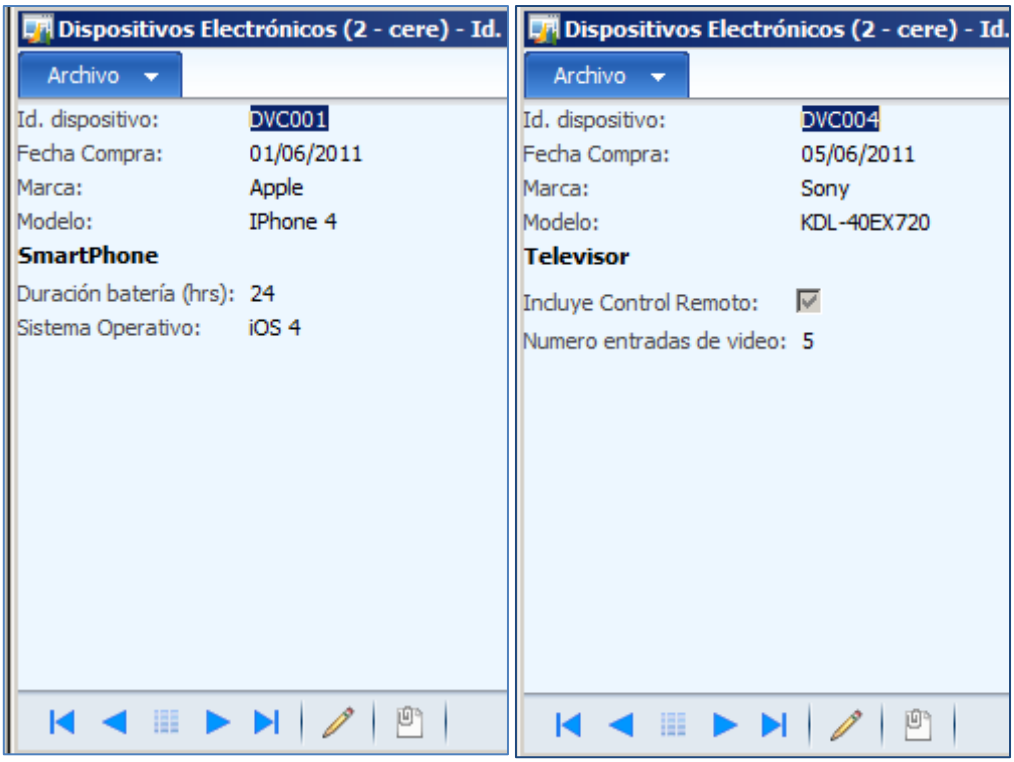
Tal como hemos visto, en la tabla de dispositivos electrónicos vamos a encontrar todos los registros, sean Smartphones o Televisores, y lo que vamos hacer será mostrar un grupo de campos u otro en función del tipo de dispositivo electrónico.

Modificamos el método "active" del datasource "DispositivoElectronico":

```
public int active()
{
    int ret;

    ret = super();

    element.lock();

    if (DispositivoElectronico is Smartphones) // Es un SmartPhone ?
    {
        GrpSmartPhone.visible(true);
        GrpTelevisor.visible(false);
    }

    if (DispositivoElectronico is Televisores) // Es un Televisor ?
    {
        GrpSmartPhone.visible(false);
        GrpTelevisor.visible(true);
    }

    element.unLock();
    return ret;
}
```

Saber si un dispositivo electrónico es un Smartphone o un televisor es tan sencillo como preguntar al sistema si la clase que representa al registro es de tipo Smartphone o de tipo Televisor (mediante la palabra reservada "**is**")

Si ejecutamos el formulario después de nuestra modificación tenemos que:

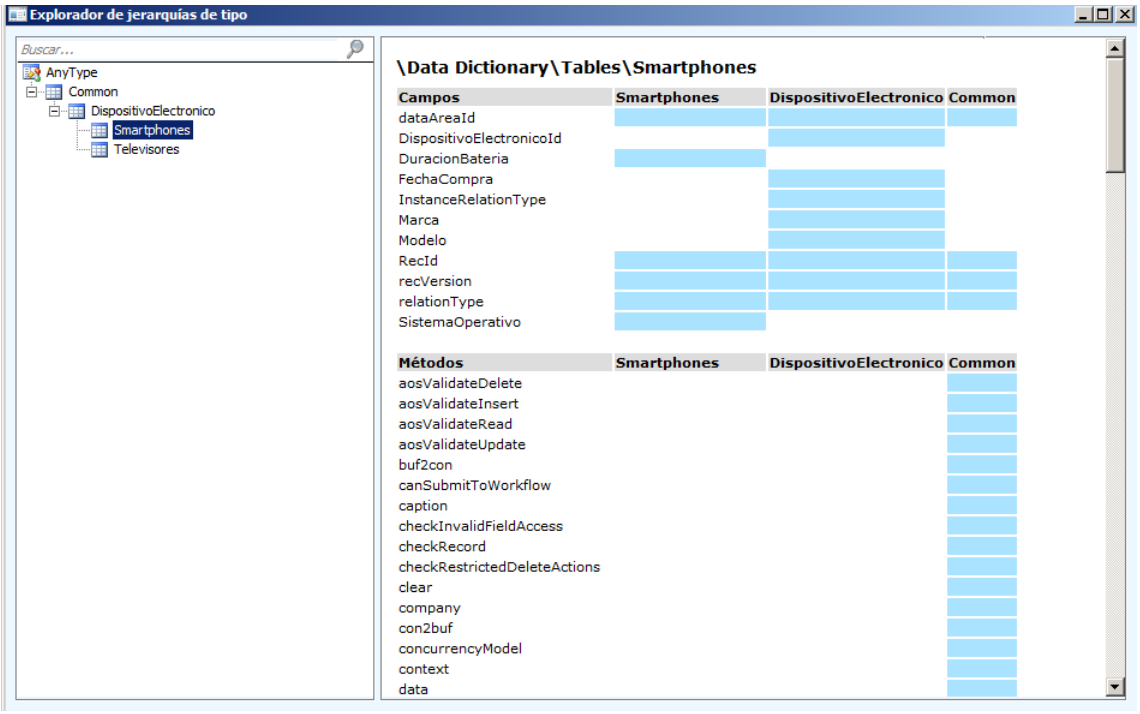


El formulario va cambiando su aspecto en función del tipo de registro que muestra.

Algunas utilidades

Disponemos de algunas utilidades que pueden ayudarnos en este nuevo paradigma de la herencia de tablas.

- Explorador de jerarquías de tipo: En el AOT, sobre un nodo de tipo Tabla, botón de la derecha del ratón, Complementos / Explorador de jerarquías de tipo. Nos muestra de forma gráfica la jerarquía de herencia y donde se implementan los distintos campos y métodos de las tablas.



- Seguimiento de instrucción SQL : Tal como hemos visto, puede sernos muy útil (sobretodo al principio) usar esta herramienta de seguimiento (no es nueva, existe desde la versión 2.5 que yo recuerde) para asegurarnos de que las consultas SQL que generamos en nuestros formularios, procesos e informes sean óptimas y no penalicen el rendimiento del sistema.

Conclusiones

Después de este ligero vistazo a una de las novedades técnicas (si, aún hay más novedades) que incorpora AX 2012, vamos a hacer un pequeño esfuerzo de reflexión para intentar determinar las ventajas e inconvenientes que puede representar en nuestro día a día.

Ventajas

- Disponemos de una alternativa muy potente de cara a realizar el diseño de los modelos de datos. Ahora, cuando afrontemos un diseño, debemos tener en cuenta la posibilidad que nos brinda el sistema de usar la herencia de tablas.
- Crear nuevas especializaciones de tablas es muy sencillo y nos permite reaprovechar código y estructuras de datos existentes. (Sería muy fácil añadir un nuevo tipo de dispositivo electrónico que representara por ejemplo a reproductores MP3 en nuestro caso práctico y aprovechar toda la funcionalidad ya creada para la tabla base).
- La gestión del modelo de clases y la programación resulta muy fácil e intuitiva.

Inconvenientes

- Un error de diseño puede llevarnos al fracaso. Esto es un hecho, con y sin herencia de tablas, pero en un modelo de datos basado en un diseño incorrecto de clases puede darse algún caso difícil de resolver ante una necesidad de “pequeños” cambios en el modelo. Ejemplos: crear un campo en un nivel de la jerarquía de tablas y luego quizá

necesitemos ese mismo campo en un nivel superior o en otra rama…, No existe herencia múltiple y podría darse el caso de que entidades de distintas ramas de la jerarquía lleguen a compartir patrones de comportamiento o estructuras de datos idénticas y… duplicar código nunca es agradable.

- Mucha precaución con el rendimiento. Tal como hemos visto, es muy importante elegir los campos en las consultas de datos. ¿Qué pasa con informes que requieran ordenaciones o agrupaciones por campos que se encuentran repartidos en diversas tablas físicas en SQL?
- Se complica la extracción/consulta de datos desde el exterior vía SQL al disponer de los datos de una misma entidad en distintas tablas y registros. (Este último punto no estoy muy seguro de que sea un inconveniente ya que siempre he sido partidario de no acceder nunca a los datos de AX vía SQL sin pasar por la capa de negocio)

¿Más ventajas e inconvenientes? Se admiten comentarios ☺