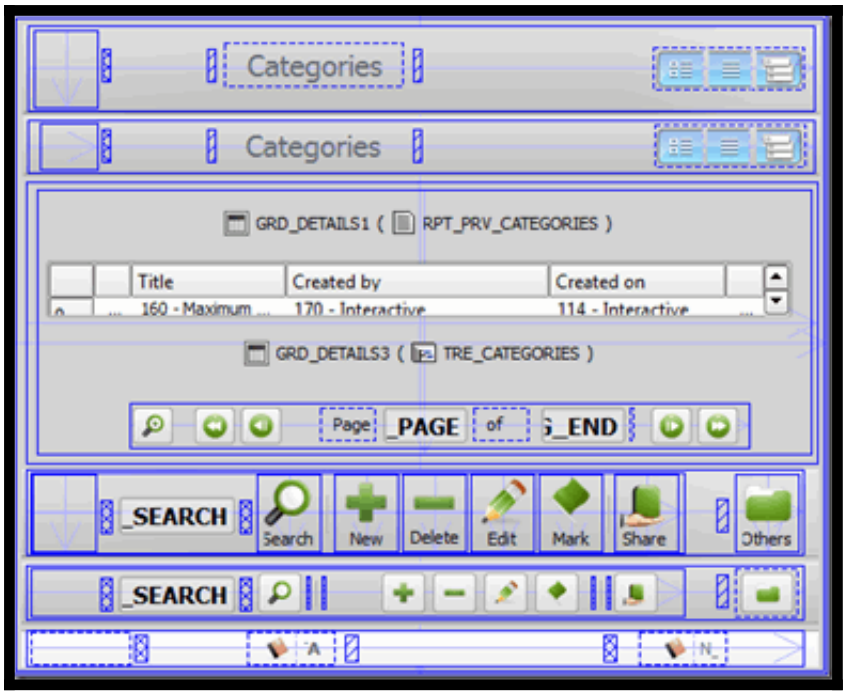
Problemas con el ciclo de vida de los formularios en PaaSOS

Una de los problemas más difíciles de resolver en la actual versión de Velneo v7 es la gestión de la carga, refresco y control de los distintos **controles objeto** ubicados en un formulario.

Si en un formulario tenemos varios **controles objeto** que muestran la misma información de distinta forma tendremos que cargar la información en todos ellos para que sea visible cuando mostremos el control adecuado. Este efecto lo conseguimos mediante las condiciones de visibilidad de los **controles objeto**.

Un ejemplo se produce en el siguiente formulario:



Si nos fijamos detenidamente vemos un formulario que contiene tres controles objeto que a su vez cargan los datos de las categorías mediante un único proceso llamado **PRC_LOAD_CATEGORIES**. Estas distintas vistas del mismo conjunto de datos permiten imprimir, visualizar en modo rejilla o en modo árbol las categorías según la configuración del módulo. Para conseguir este efecto condicionamos la visualización de cada uno de los **controles objeto** en función de la propiedad **condición de visibilidad** asociada a una variable local del formulario rellenada en el evento **starting** del formulario en base a la configuración del módulo almacenada en una tabla de configuración**.** El primer control tendría asociada la condición “DEFAULT_VIEW=1”, el segundo “DEFAULT_VIEW=2” y el tercero “DEFAULT_VIEW=3”.

Al cargar el formulario por primera vez los datos se leen tantas veces como llamadas producidas por los tres controles objeto. **Esto implica cargar tres veces el mismo conjunto**

de datos. Lógicamente esta carga de datos triplicada implica una **perdida importante de rendimiento** (especialmente en SaaS). Este comportamiento de los controles objeto implica cargar una información en tres controles que no será visualizada, con el correspondiente consumo de tráfico de red y procesamiento asociado en cliente/servidor.

> Lo lógico sería que el **control objeto** no se cargase si la **condición de actividad** del mismo estuviera deshabilitada.

Además del problema de la carga de datos triplicada tenemos otra cuestión fundamental que resolver, ya que **el evento starting** del formulario (donde leemos el valor de la configuración de nuestro módulo) **se lanza posteriormente a la carga de los tres controles objeto**. esta situación provoca que los controles objeto no se comporten como esperamos.

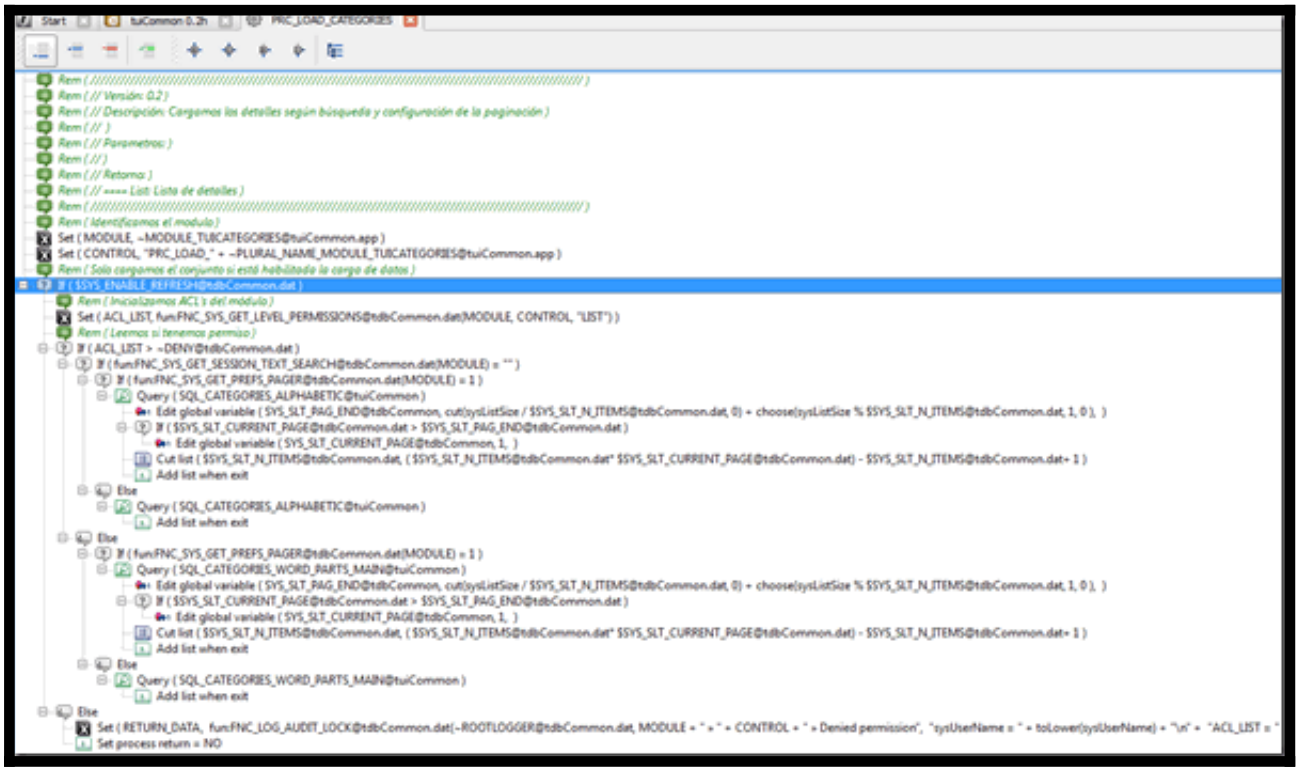
> Lo ideal sería que el evento **starting** del formulario se produjese en el inicio del mismo, antes de la carga de los controles.

La solución:

La solución pasa por tener un absoluto control de la carga de datos.

Si en el proceso de carga de los datos solo retornamos el conjunto una sola vez conseguiremos ser óptimos en los consumos de ancho de banda y en procesamiento.

Para resolver esta cuestión nosotros hemos utilizado una variable global que gestiona el control de la carga de datos:



Esta variable toma valor true solo cuando es estrictamente necesario, por lo tanto, solo se carga el conjunto de datos adecuado si previamente se habilitó la variable. En nuestro caso la variable esta siempre a false excepto cuando refrescamos el **control objeto** que deseamos visualizar.

Este refresco se produce **en el evento starting** del formulario **y en el evento de refresco** que se lanza al cambiar de visualización.

Con este código refrescamos solo el control adecuado en función de la configuración del modulo y de la vista seleccionada. Previamente tenemos que habilitar el refresco y cuando hemos terminado tenemos que volverlo a deshabilitar.

> Este montaje solo carga el conjunto una sola vez en el evento **starting** y otra cuando cambiamos el modo de visualización mediante el evento **refresh**.

Espero que este ejemplo os ayude a mejorar el rendimiento de vuestras aplicaciones en la nube. En **PaaSOS**, esta solución en combinación con otras han mejorado el rendimiento considerablemente.