

El infierno (y IV) de las variables globales en Velneo V7

Nos comenta Sergi en el foro de V7 que tiene un problema con las variables globales... jejeje y yo... <http://velneo.es/foros/topic/variables-globales-panos-de-ejecucion-y-demas/>

Intentaré resolver sus dudas, aunque creo que Sergi lo tiene bastante claro (en el foro no puedo escribir, me spammea... David no se te olvide mirarlo):



[\[N1\] sergi.esteve](#)

Tenía hecho un sistema de permisos, por el cual en los triggers de la bbdd comprobaba si el usuario actual tenía permiso para hacer alta , modificación o borrado en esa tabla.

Funcionaba perfectamente, ya que usaba la variable sysUserName para saber el nombre de usuario y consultaba una tabla de permisos.

Se me ha ocurrido hacer una tabla de usuarios y hacer un login personalizado, **cuando entras te logas y se guarda en una variable global el usuario con el que has entrado.**

Cambié la función que comprueba los permisos de sysUserName a dicha variable global en memoria.

Por supuesto no funciona, la variable en el server no tiene valor.

No veo cómo conseguir que el servidor sepa con qué usuario has entrado.

He probado ha hacer una tabla en memoria de sesiones y cargar en tercer plano un registro, pero una vez insertado ya no sé cómo recuperarlo.

He buscado alguna variable o función javascript del tipo sysUserName pero que me devuelva un id de sesión o algo para poder usar una tabla de sesiones con usuarios o algo, pero no he encontrado nada.

¿Cómo puedo enfocararlo para conseguirlo? ¿alguien se ha visto en situación similar?

Gracias por adelantado y perdonar el tostón.

Un poco de teoría....

Comportamiento de las variables globales

Las variables globales tienen un comportamiento claramente definido en el siguiente artículo de V7;

<http://velneo.es/ambito-de-las-variables/>

Resumiendo:

Ambito de las variables Globales

El contenido de una **variable global en disco** es común a todos los usuarios y [planos de ejecución](#) (tanto en los clientes como en el servidor).

El funcionamiento de las **variables globales con persistencia en memoria** se circunscribe al ámbito estándar de funcionalidad de este tipo de variables en los lenguajes de programación genéricos que, básicamente, consiste en ser globales a la máquina en la que se haga uso de ellas.

A la práctica...

Soluciones

Ok... ya sabemos... entonces ¿Cómo hago para guardar y usar el ID del usuario que accedió a nuestra aplicación?

Soluciones:

- **sysUserName**: funciona ¿pero y si quiero hacer un login propio independiente del de V7?
- **Variable global**: no vale ya que si la pones en memoria no será visible en 3P y si la pones en disco tampoco funcionaría.
- **En disco mediante una tabla**: Es una solución funcional en 1, 2 y 3er plano pero hay que tener cuidado con los consumos en el vServer.

La nuestra...

Nuestra solución

Solución aportada en PaaSOS para esta cuestión... **En disco mediante una tabla.**

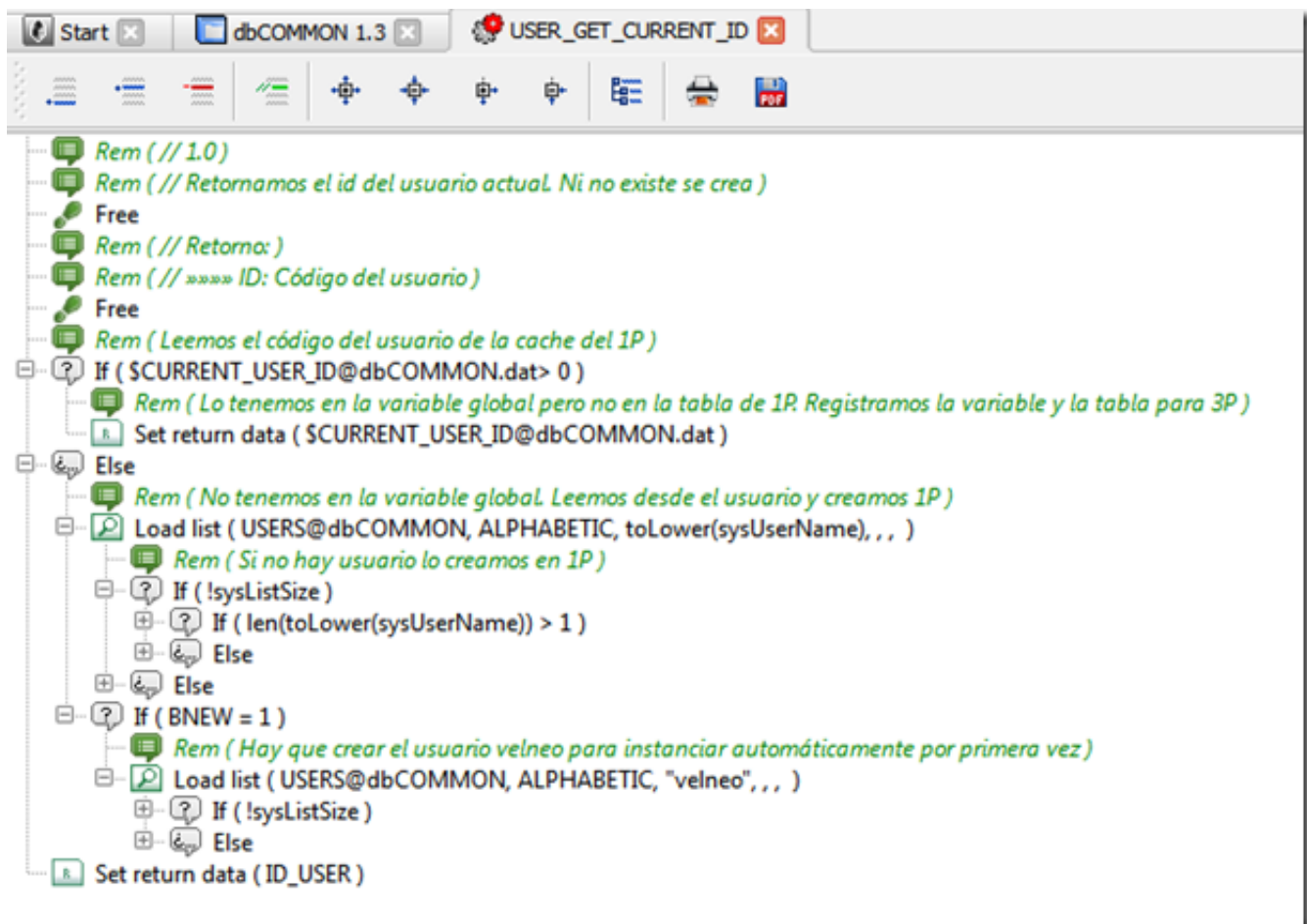
1.- Guardaremos los usuarios en la tabla (USERS)

2.- Disponer de dos funciones para obtener el ID del usuario que está logueado (el ID en la tabla USERS):

- USER_GET_CURRENT_ID (Para 1er y 2º plano)
- USER_GET_CURRENT_ID_3P (Para 3er plano)

En ambas podemos usar sysUserName sin miedo ya que retorna correctamente el usuario en curso y localizar el ID en la tabla USERS en base al nombre del usuario en el vServer.

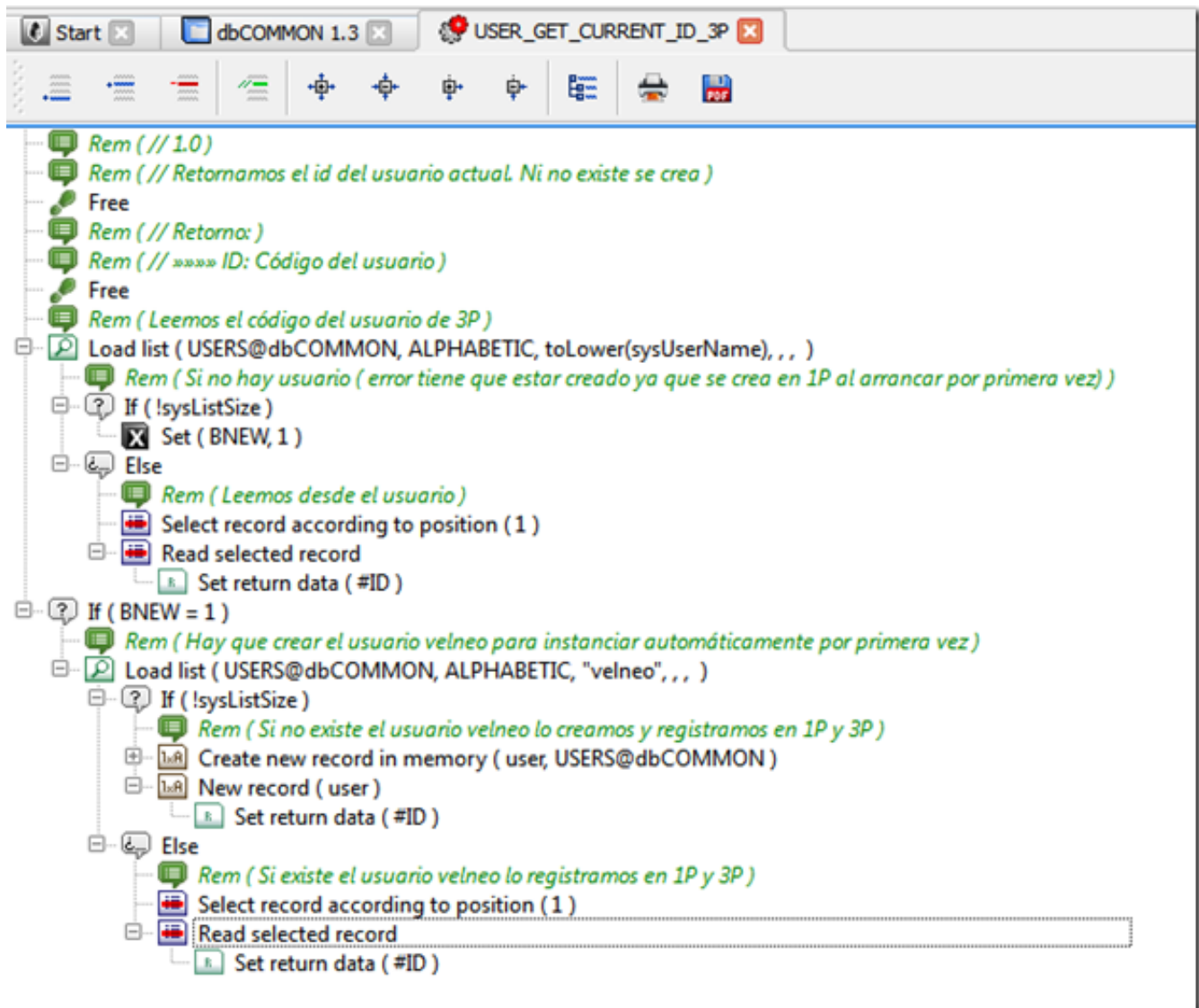
Hay que tener en cuenta que si en ejecuciones de 1 y 2 plano requeriremos de ser lo más óptimos en los accesos a disco (evitar llamadas a servidor). Para ello, lo mejor una variable local en memoria (visible en 1 y 2 plano).



```

Rem ( // 1.0 )
Rem ( // Retornamos el id del usuario actual. Ni no existe se crea )
Free
Rem ( // Retorno: )
Rem ( // >>>>> ID: Código del usuario )
Free
Rem ( Leemos el código del usuario de la cache del 1P )
If ( $CURRENT_USER_ID@dbCOMMON.dat > 0 )
  Rem ( Lo tenemos en la variable global pero no en la tabla de 1P. Registramos la variable y la tabla para 3P )
  Set return data ( $CURRENT_USER_ID@dbCOMMON.dat )
Else
  Rem ( No tenemos en la variable global. Leemos desde el usuario y creamos 1P )
  Load list ( USERS@dbCOMMON, ALPHABETIC, toLower(sysUserName), , , )
  Rem ( Si no hay usuario lo creamos en 1P )
  If ( !sysListSize )
    If ( len(toLower(sysUserName)) > 1 )
      Else
    Else
  If ( BNEW = 1 )
    Rem ( Hay que crear el usuario velneo para instanciar automáticamente por primera vez )
    Load list ( USERS@dbCOMMON, ALPHABETIC, "velneo", , , )
    If ( !sysListSize )
      Else
  Set return data ( ID_USER )
  
```

En 3P ese problema es menor ya que no tenemos problemas de roundtrip (ya estamos en el servidor por lo que no es casi costoso añadir una consulta por índice).



También hay que tener en cuenta el comportamiento en los procesos ON_INIT_SERVER ya que no hay usuario logueado aún... nosotros, todas las operaciones lanzadas sin usuario autenticado las asociamos a la cuenta de servicio "velneo"

Bueno... como veis el tema de las variables globales es un pequeño infierno al que tenéis que dar solución para cuestiones como estas. Pero el problema fundamental no solo es en esta situación, hay otras muchas donde las variables globales se quedan cojas. Nosotros tenemos un subsistema llamado "PERSISTENCIA" para dar soluciones a cada uno de los problemas que puedas necesitar resolver.

De esto y otras cosas hablaremos pronto en el primer curso de PaaSOS para programadores que celebraremos la semana que viene.



Un Saludo a todos los Velneadores