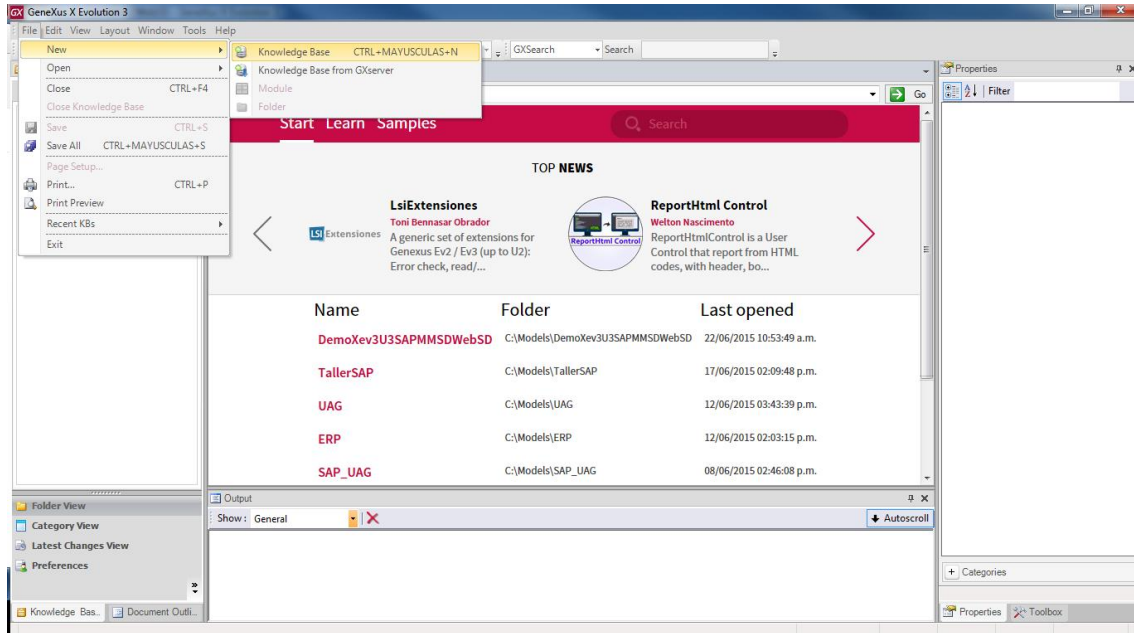
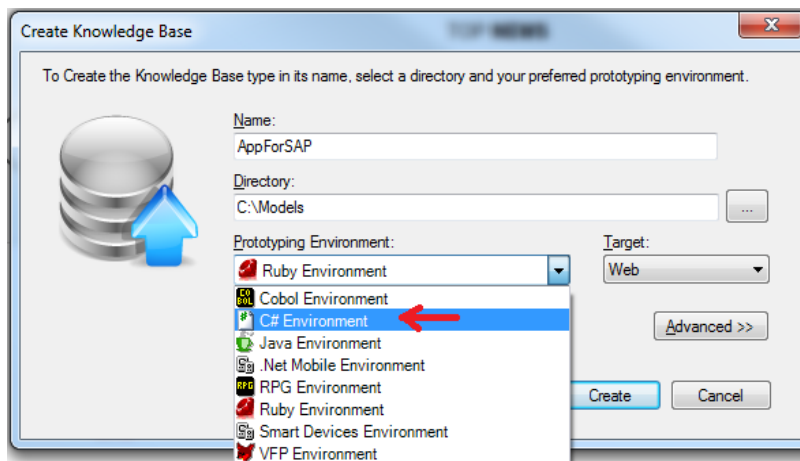


Creando una aplicación móvil con GeneXus para trabajar con los Materiales del SAP ERP – parte 1

Empezamos por crear una base de conocimiento (knowledge base: KB). La base de conocimiento corresponde al proyecto de la aplicación.



Aquí especificamos su nombre, y aquí el ambiente de desarrollo con el que prototiparemos la aplicación. Elegiremos C#.



Al presionar **Create** empezarán a importarse todos los objetos necesarios para inicializar la base de conocimiento.

En el Folder View aparecerán, entre otras cosas, todos los objetos GeneXus que implementarán nuestra aplicación.

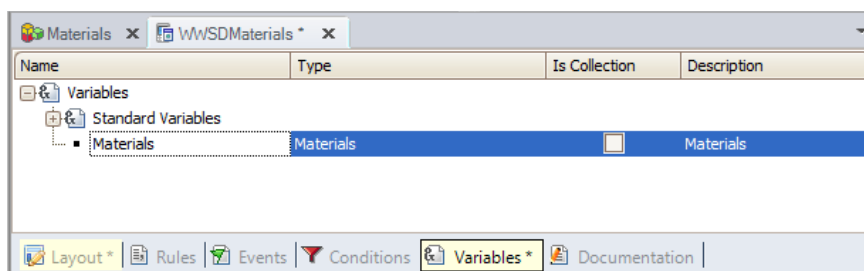
Empezaremos por implementar el panel para mostrar los Materiales del ERP.

De cada material nos interesará mostrar el identificador y la descripción, por lo que nos crearemos un tipo de datos estructurado al que llamaremos Materials, que será una colección de ítems a los que llamaremos Material, cada uno de los cuales estará conformado por un MaterialId de tipo Character de 18 y de un MaterialDescription de tipo Character de 40.

Creemos, ahora sí, el panel para mostrar los materiales: filtremos por objetos para Smart Devices → Panel for Smart Devices. Lo llamaremos WW (por Work With) SD (por Smart Devices) Materials.

Vemos que hay varias solapas para implementar las distintas partes del objeto.

Empezaremos por definir una variable del tipo de datos estructurado que acabamos de definir. Para ello arrastramos el tipo de datos y nos queda definida una variable con el mismo nombre.



Querremos mostrar su contenido en la pantalla del panel. Lo que hacemos, entonces, es insertar la variable como un control. Podríamos elegir mostrar solamente las descripciones, pero no, mostraremos todo. Vemos que automáticamente GeneXus creó un control Grid, con dos controles de tipo variable: uno para el identificador del material y el otro para la descripción.

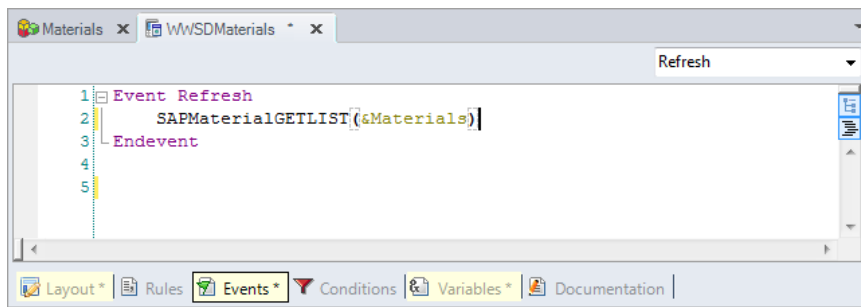
Al estar basado en una variable estructurada GeneXus cargará este grid automáticamente en ejecución con lo que contenga la variable en ese momento.

Podemos agregar controles dentro del grid, por ejemplo una tabla para alinear los controles entre sí ... Colocamos dentro los controles. Al identificador de material le habilitamos la etiqueta ubicada a la izquierda, y le cambiamos su caption por ID. Podemos agregar imágenes que se hayan importado en la base de conocimiento por ejemplo esta, para que el usuario pueda luego hacer tap sobre ella y actualizar los datos del material elegido. Y lo mismo para poder eliminar. Aquí ya hemos modificado las propiedades de las filas y columnas y de los controles, para que queden alineados, para que ocupen los espacios que nos interesan, etcétera. No nos detendremos en estos detalles. Vea que en la pestaña Style se le muestra esquemáticamente cómo lucirá el layout en ejecución.

Sólo nos resta cargar la variable Materials. La programación de los paneles se realiza a través de eventos. Necesitamos cargar la variable antes de que se ejecute la carga del grid en pantalla. Ese momento es el evento del sistema Refresh, que se produce antes de cargar la parte fija del panel. Inmediatamente después de este evento se realizará la carga del grid en pantalla de acuerdo al contenido de la variable.

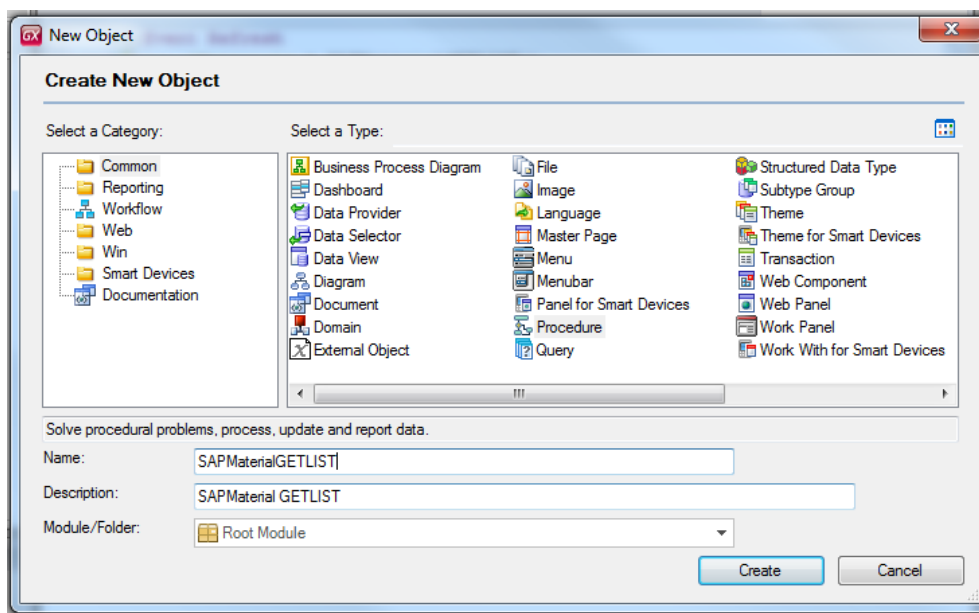
Aquí tenemos que realizar la carga de la variable.

Para ello llamamos a un procedimiento GeneXus, al que llamaremos..., al que le pasaremos la variable Materials (las variables se escriben prefijadas con el carácter ampersand) para que la devuelva cargada.



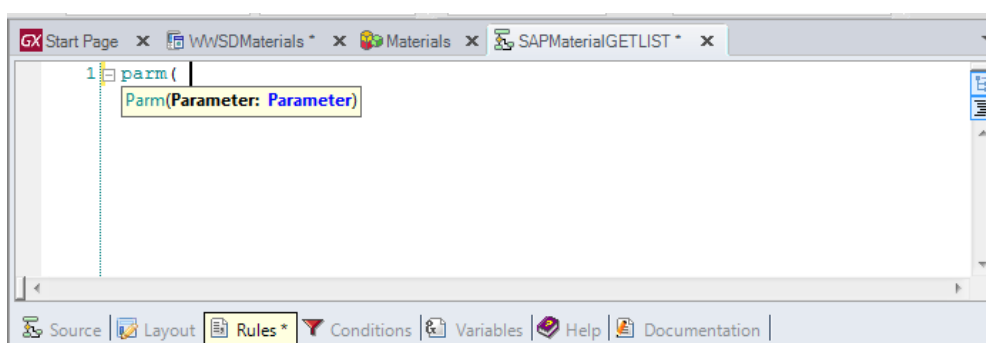
Este procedimiento será el que implementará la llamada RFC a la BAPI correspondiente.

Tenemos que crearlo:

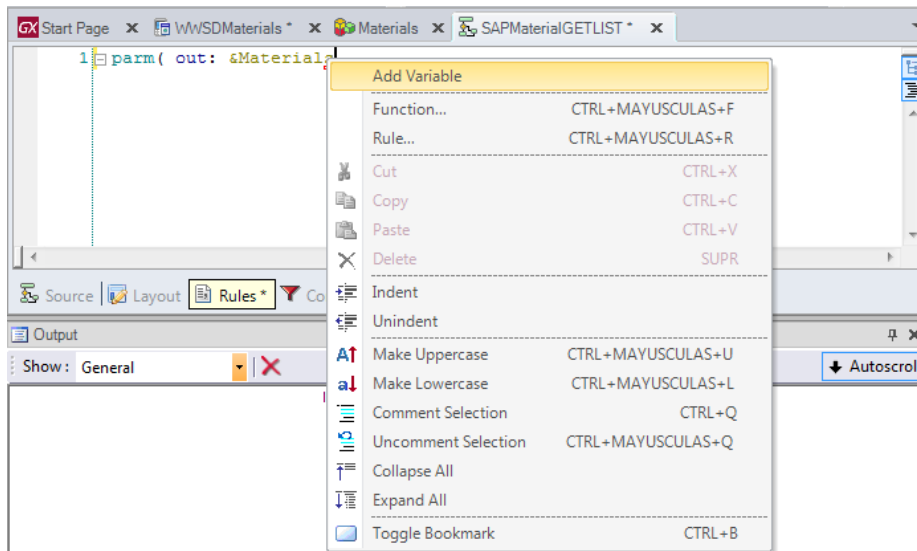


Este tipo de objeto es excepción en GeneXus en cuanto a que su lógica es casi completamente procedural, al contrario de lo que sucede con la tendencia general de GeneXus, que va en la línea de lo declarativo.

Primero indiquemos que devolverá una lista de materiales como resultado de su ejecución. Para ello, en la sección de reglas, especificamos la regla parm, que declara los parámetros de entrada y de salida.



En principio sabemos que habrá un parámetro de salida, que será una variable que también llamaremos &Materials. La agregamos como variable:



Como la llamamos igual que un tipo de datos, lo asume por defecto.

Ahora en el Source debemos cargarla a través de la BAPI. Para ello, debemos importar esa BAPI. ¿Cómo lo hacemos?

Lo veremos en el próximo video...

