

## DESAFÍO 2:

Para este desafío comenzaremos importando el *xpz* de los insumos que les proporcionamos.

Dentro de los elementos para importar, hay una transacción CustomUser que va a tener usuarios personalizados contra los que GAM va a chequear para verificar identidades. Vamos a tener un procedimiento CustomUserLoad que se va a encargar de poblar la transacción CustomUser con usuarios por defecto. La tabla asociada con esa transacción. Van a haber *SDTs* auxiliares (estos sirven para implementar el procedimiento de autenticación Custom). Y dominios y carpetas que sirven nada más para una estructura más organizada y centralizada de los elementos a importar.

Una vez importados los objetos, procedemos a crear el procedimiento auxiliar de autenticación Custom.

Este procedimiento será Main, y el código luce de la siguiente manera.

Antes de profundizar en éste, agregamos las reglas donde tenemos un string de entrada de tipo VarChar, y uno de salida también VarChar.

Ahora si en el código, necesitamos una variable *Key* de tipo VarChar que estará definida en el GAM para el tipo de autenticación Custom, la cual mostraremos más adelante.

Luego hay que utilizar los SDT auxiliares que se importaron, para las distintas variables que se van cargando.

Después tenemos strings asociados al nombre de usuario y contraseña, también VarChar, donde vemos que se descifran desde el string de entrada que es un JSON utilizando la key, lo cual así lo establece GAM.

Luego se carga un SDT, que tendrá la salida de ejecución de este procedimiento.

La versión de JSON de GAM, para este caso será 2.0, donde también funciona con la 1.0. Solo habría que adaptar el ejemplo a ésta.

Finalmente se ejecuta la subrutina *ValidUser* que validará los parámetros de entrada que son el UserLogin y UserPassword contra la tabla CustomUser que importamos, comparando los campos, y como un extra, verificando que el usuario esté activo. Si todas las condiciones cumplen, se van a cargar todos los datos del usuario en el SDT de salida que luego GAM gestionará. En caso contrario, si no se cumplen todas las condiciones, se devuelve un error en el Status que proporciona GAM.

El siguiente paso es crear el tipo de autenticación Custom en el Backoffice web de GAM. Para eso nos logueamos con el usuario administrador, settings, authentication types, y presionamos en Add.

Seleccionamos el tipo Custom.

Como nombre, le pondremos "Custom login". La función, "solo autenticación". Va a estar habilitado. La versión de JSON como dijimos era la 2.0. Acá generamos la clave que comentábamos hoy, la cual introduciremos en nuestro procedimiento creado posteriormente.

Como nombre de archivo para este caso de .NET, el nombre será esta dll (comienza con “a” porque así comienzan los procedimientos Main en estas versiones de GeneXus). El Package, tendrá el valor “GeneXus.Program” (que es el namespace que englobaría la clase del procedimiento). La clase que aplica en este caso es la misma que el *File name* pero sin el “.dll”.

Copiamos la clave y confirmamos.

Ahora en GeneXus, pegamos la clave copiada en la variable Key, y el siguiente paso es hacer un build para que se apliquen todos los cambios.

Como se tiene la nueva tabla CustomUser, es necesario reorganizar la base de datos para crearla. Este proceso demora unos minutos, así que avanzamos rápidamente.

Ya finalizado este proceso, hay que cargar los usuarios en la tabla CustomUser y para eso utilizamos el procedimiento entregado, que podemos ejecutar sin Building porque ya se hizo antes el Build All.

Una vez terminado eso, ya podemos dirigirnos al login de la aplicación donde vemos que tenemos para elegir mas de un tipo de autenticación: el por defecto, y custom login que es el nuevo.

Usaremos uno de los usuarios cargados (que es Mick) con contraseña del 1 al 6.

Vemos que el login es exitoso, pero que luego de seleccionar dirigirnos al Backoffice de GAM nos devuelve no autorizado. Esto pasa debido a que el usuario Mick no tiene permisos para ver el backoffice, pero es solo un tema de permisos. El desafío quedó completado.