

# GAM

Update to GeneXus  
from Evolution 3 to version 15

*GeneXus™ 15*

## Overview

- Backend: Smooth and RWD
- Dynamic Permissions Menus
- Events subscription
- Logout options for Single Sign on using GAM
- Support for SAP Hana database

El GAM cuenta con un nuevo Backend que es responsive y smooth (y usa el tema Carmine).

Con respecto a nuevas funcionalidades en esta versión tenemos:

- La posibilidad de definir menús dinámicos en GAM (<http://wiki.genexus.com/commwiki/servlet/wiki?32336>.)
- La posibilidad de suscribirnos a eventos generados por el GAM (por ejemplo cuando se crea un usuario en GAM queremos que se dispare cierta acción como puede ser un mail de bienvenida, etc).
- Se mejoraron las opciones de logout en SSO

Por otro lado esta versión cuenta con soporte para SAP.

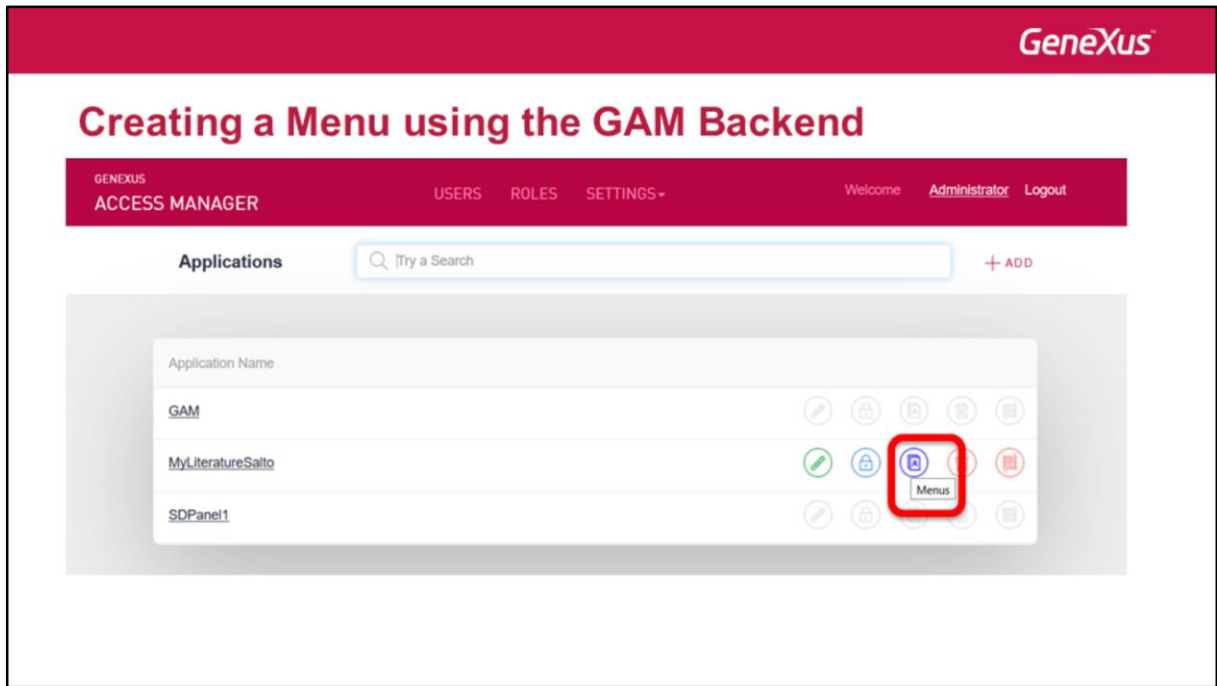
# Backend Overview

## **DEMO**

*GeneXus™ 15*

# Dynamic Permissions Menus

*GeneXus™ 15*

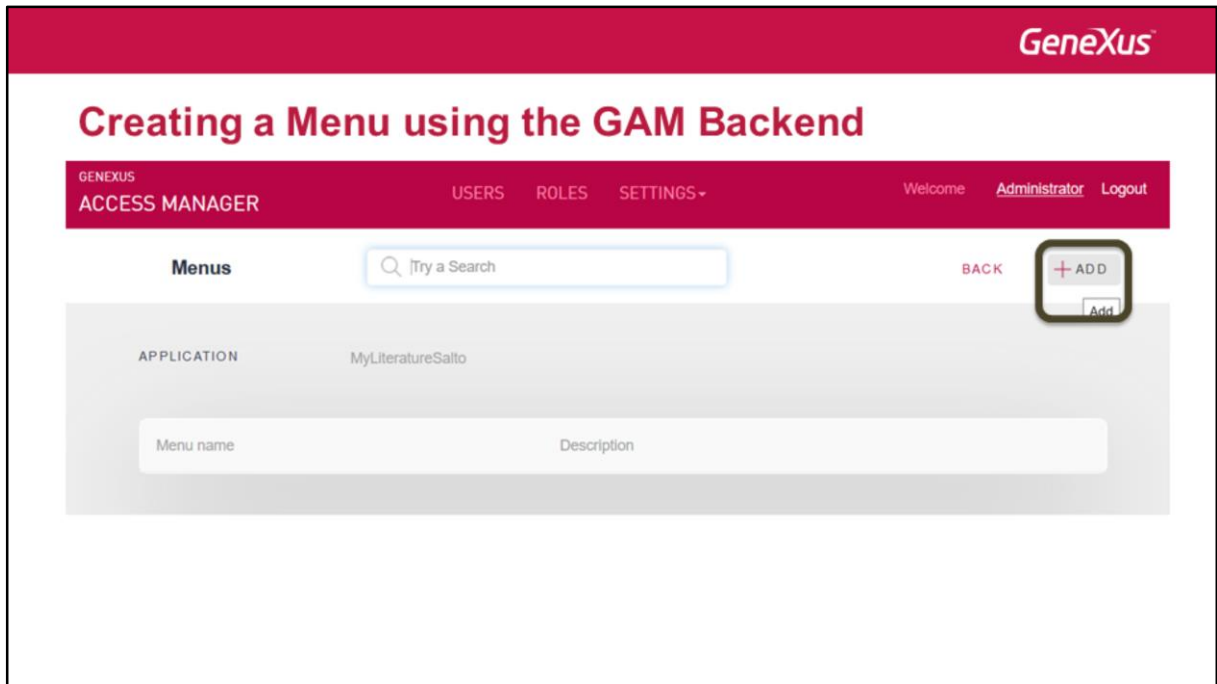


Ahora podemos crear menús dinámicos en run-time, que serán visibles o no según los roles (y sus respectivos permisos) del usuario logueado.

GAM va a retornar una estructura con los menús disponibles para el usuario logueado, esta estructura puede ser cargada en run-time por cualquier UserControl (o un menú desarrollado en la aplicación).

Por mas información ver:

<http://wiki.genexus.com/commwiki/servlet/wiki?29681,Howto%3A%20Define%20a%20Menu%20using%20GAM>



Los menús se pueden dar de alta en el backend del GAM de forma interactiva o se pueden dar de alta programáticamente en un proceso de inicialización vía la API.

Las siguientes pantallas nos muestran cómo hacerlo desde el backend.

Comenzamos haciendo clic en 'Add Menu'.

## Creating a Menu using the GAM Backend

GENEXUS  
ACCESS MANAGER

USERS ROLES SETTINGS ▾

Welcome [Administrator](#) [Logout](#)

**Menu** ✓ CONFIRM ✕ CANCEL

Application	MyLiteratureSalto
GUID	<input type="text"/>
Name	AppBackend
Description	App Backend Menu

Definir el menú dándole una descripción.

## Creating a Menu using the GAM Backend

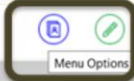
GENEXUS  
ACCESS MANAGER

USERS ROLES SETTINGS ▾

Welcome [Administrator](#) [Logout](#)

**Menus**  [BACK](#) [+ ADD](#)

APPLICATION	MyLiteratureSalto
Menu name	Description
<a href="#">AppBackend</a>	App Backend Menu

 Menu Options

Se pueden agregar opciones al menú.



## Creating a Menu using the GAM Backend

The screenshot shows the 'MenuOption' form in the GeneXus ACCESS MANAGER. The form is titled 'MenuOption' and has a 'CONFIRM' button. The form fields are:

- Application
- Menu
- GUID
- Name
- Description
- Option Type
- Resource
- Resource parameters

The 'Resource' field is open, showing a list of permissions. The list includes:

- autores\_Delete (autores)
- autores\_Execute (autores)** (highlighted in blue)
- autores\_FullControl (autores)
- autores\_Insert (autores)
- autores\_Services\_Delete (autores)
- autores\_Services\_Execute (autores)
- autores\_Services\_FullControl (autores)
- autores\_Services\_Insert (autores)
- autores\_Services\_Update (autores)** (highlighted in green)
- autores\_Update (autores)
- gameexampleapplicationentry\_Execute (gameexampleapplicationentry)
- gameexampleappmenuentry\_Execute (gameexampleappmenuentry)
- gameexampleappmenuoptionentry\_Execute (gameexampleappmenuoptionentry)
- gameexampleapppermissionchildren\_Execute (gameexampleapppermissionchildren)
- gameexampleapppermissionentry\_Execute (gameexampleapppermissionentry)
- gameexampleapppermissionselect\_Execute (gameexampleapppermissionselect)
- gameexampleauthenticationtypeentry\_Execute (gameexampleauthenticationtypeentry)
- gameexamplechangerepository\_Execute (gameexamplechangerepository)
- gameexamplechangeyourpassword\_Execute (gameexamplechangeyourpassword)
- gameexampleconnectionentry\_Execute (gameexampleconnectionentry)

The 'Resource' field is currently set to 'autores\_Execute (autores)'.

Cuando se agrega una opción de menú vamos a poder seleccionar si la opción es Simple o Menú (para submenús).

Cada opción de menú está asociada a un Recurso – que podemos seleccionar desde un combo – que muestra los permisos definidos en la aplicación.

El menú que se dibuja en la aplicación va a tener solo las opciones para las cuales el usuario logueado tiene acceso (según los Roles del usuario logueado y los permisos asociados a dichos roles).

Los recursos se definen automáticamente (son los permisos generados por GeneXus), pero también podemos definir manualmente recursos (manualmente o vía la API).

## Example: getting the menu options

```
for &AppMenuOption in &Application.GetMenuOptions(&MenuId,  
&Filter, &Errors)  
  
    &Id      = &AppMenuOption.Id  
    &Name    = &AppMenuOption.Name  
    &Dsc     = &AppMenuOption.Description  
    &Type    = &AppMenuOption.Type  
endfor
```

A la hora de desplegar el menú en la aplicación, se puede utilizar cualquier UserControl (ver <http://wiki.genexus.com/commwiki/servlet/wiki?29743,Howto%3A+Display+a+GAM+Menu+using+Jscoockmenu+UC,> )

# Event Subscription

*GeneXus™ 15*

## Event Subscription

- **User\_Insert**
- **User\_Update**
- **User\_Delete**
- **User\_UpdateRoles**
- **Role\_Insert**
- **Role\_Update**
- **Role\_Delete**
- **Repository\_Login**
- **Repository\_Logout**

El objetivo de estos eventos es permitir extender la funcionalidad de GAM, permitiendo ejecutar código externo a GAM cuando alguno de estos eventos es ejecutado.

El poder suscribirse a eventos hace que el código sea más limpio/mantenible. Consideremos un escenario donde la aplicación mantiene una tabla de usuarios paralela a la de GAM. Siempre que interactuamos con GAM para insertar, modificar o borrar usuarios tendríamos que tener el código que mantiene las tablas sincronizadas. Por ejemplo al grabar datos de un usuario GAM tendríamos

```
&GAMUser.Save()  
UpdateLocalUser(&GAMUser) // procedimiento que mantiene la redundancia local
```

Y tendríamos que asegurarnos que siempre estas llamadas se hacen juntas en todo el código en donde se actualice el usuario GAM.

La suscripción al evento `User_Update` evita que tengamos que mantener este código a mano, y nos asegura que siempre que se ejecuta el evento `update` de un usuario se va a ejecutar el código de nuestra aplicación que está suscripto al evento.

Por más información:

<http://wiki.genexus.com/commwiki/servlet/wiki?32698,GAM%20Events%20subscription>

## Logout Options

*GeneXus™ 15*

## SSO Logout Options

- Client Side Only
- **New Options:**
  - Identity Provider & Client
  - Identity Provider & All-Clients

Para GeneXus Evolution 3 y versiones previas, el método `GAMRepository.Logout()` realizaba un logout local en el cliente.

Desde GeneXus 15 tenemos dos nuevas opciones para configurar ( por más información ver [Logout options for Single Sign On using GAM](#)

[Note](#))

. **Client Side Only** – Esta era la única opción existente hasta ahora. El logout sucede en el cliente local pero en el servidor la sesión continua siendo válida y siguen conectadas las demás aplicaciones que el usuario tenga activas. Mientras la sesión en el servidor de Identidades siga activa, podría volver a conectarme sin tener que loguearme

. **Identity Provider & Client** – Esto nos desloguea del cliente local y también del servidor de identidad, pero otras sesiones que el usuario tenga en otros clientes siguen conectados mientras no tengan que ir al servidor. Al ir al servidor de identidades se pide login nuevamente.

. **Identity Provider & All Clients** – Esto nos desloguea del cliente local y todos los clientes que el usuario tenga activos. También del servidor de identidad, es decir, cualquier interacción del usuario en cualquier cliente va a necesitar un nuevo login.



Videos

[training.genexus.com](https://training.genexus.com)

Documentation

[wiki.genexus.com](https://wiki.genexus.com)

Certifications

[training.genexus.com/certifications](https://training.genexus.com/certifications)