

## Applying And Using GAM



Una funcionalidad que vemos cada vez más en las aplicaciones web, es la posibilidad de autenticarse una sola vez para todas las aplicaciones que estamos usando.

El **Single Sign On** (o autenticación única) es el procedimiento mediante el cual un usuario se autentica en una aplicación WEB y ya queda autenticado en el resto de las aplicaciones que ejecuten en distintas solapas de la misma ventana del browser.

Esto puede hacerse debido a que una aplicación con GAM puede ser proveedora de identidad, como lo hace Facebook o lo hace Google+ y otras aplicaciones con GAM pueden autenticarse a través de la misma.



Con Single Sign On (SSO), si todas estas aplicaciones se ejecutan en una misma instancia del browser (en diferentes tabs), alcanza con loguearse una única vez.

Todo esto es configurable en tiempo de instalación, no requiere programación, solo configuración del módulo de seguridad del GAM.

El mecanismo de Single Sign On usado en GeneXus se basa en el estándar Oauth 2.0.

Si estoy logueado por Single Sign on, significa que en un mismo browser estoy ejecutando varias aplicaciones en distintos tabs, todos en la misma sesión del browser. Pero ahora, puedo

necesitar desloguearme, y puede pasar que quiera desloguearme de una sola aplicación o de varias aplicaciones. Para eso, tengo básicamente, tres situaciones...

ENTERPRISE | SecurityGeneXus |

### Single Sign On - Logout

- Client Side Only
- Identity Provider & Client
- Identity Provider & All Clients

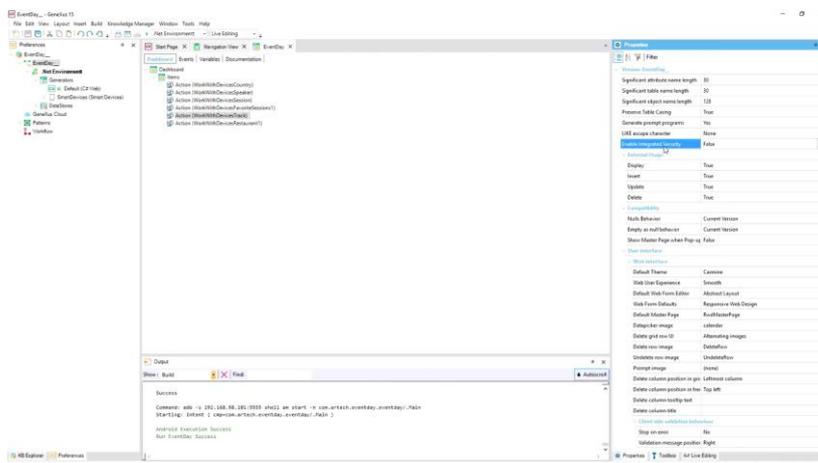


GeneXus Access Manager

More info: <http://wiki.genexus.com/commwiki/servlet/wiki?25385>

- **Client Side Only** – Me deslogueo de una aplicación y el resto de las aplicaciones siguen conectadas al proveedor de identidades. Quiere decir que solamente una aplicación es la que se va a desloguear.
- **Identity Provider & Client** – Se finaliza la sesión en el cliente y también en el proveedor de identidades. El resto de las aplicaciones van a estar vivas mientras su sesión local esté viva, pero cuando se requiera se va a pedir login de nuevo.
- **Identity Provider & All Clients** – En este caso se finaliza la sesión del server y también de todos los clientes, y en este caso, las aplicaciones tendrán que loguearse nuevamente para poder ingresar.

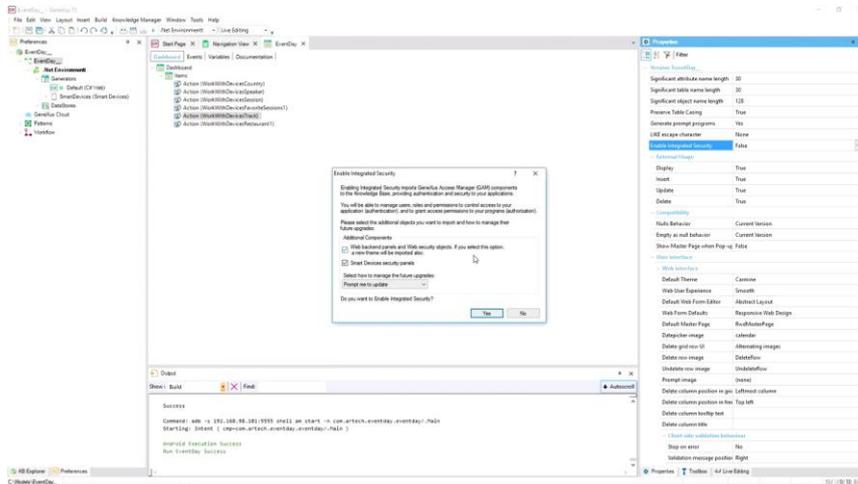
A continuación vamos a realizar una demo básica de cómo podemos agregar autenticación a una aplicación hecha con GeneXus. Veremos lo sencillo que es realizar este tipo de tarea, cambiando solamente unas propiedades en la versión activa. Así que para eso, vamos a GeneXus.



Aquí estamos en la ventana de Preferences, y a nivel de la versión activa, tenemos la propiedad **Enable Integrated Security** en False. Vamos a activar esto: le vamos a poner True.

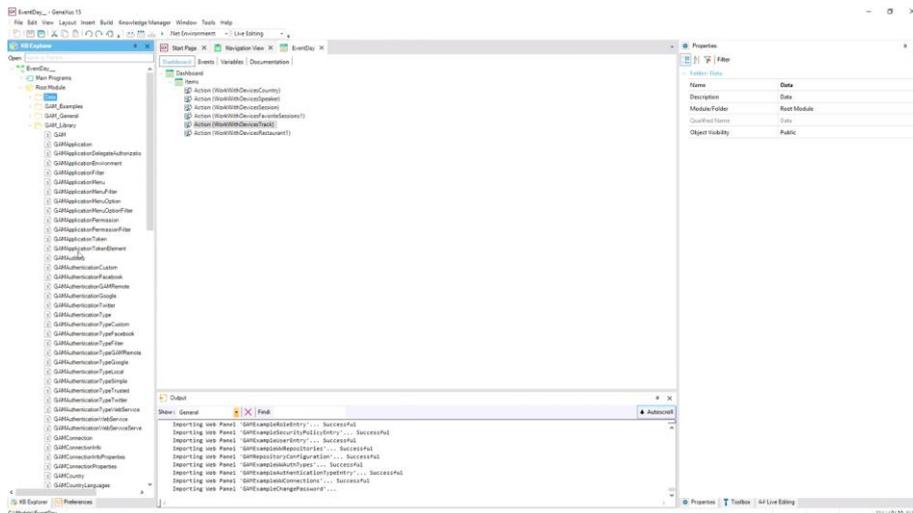
Y ven que se abre un cuadro de diálogo, donde me sugiere que va a instalarse todo el módulo de seguridad del GAM en mi KB.

Recordemos que son objetos GeneXus, que fueron desarrollados con GeneXus, que se van a integrar a mi KB, y voy a tener además disponible un montón de objetos externos que son la API que voy a poder utilizar para poder acceder programáticamente a distintas funciones del GAM.



Así que lo que voy a hacer ahora es darle “Yes”... voy a dejar **Integrated Security Level** en el valor Authentication, porque solamente me interesa hacer la autenticación, y vemos que se están importando un montón de objetos externos, dominios, y otros recursos del GAM, que son aquellos objetos de la KB del GAM que se incorporan a mi KB.

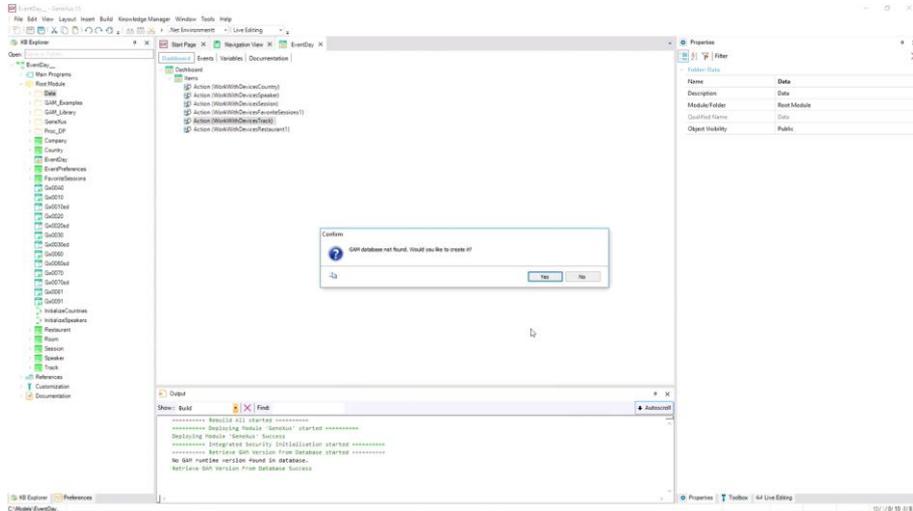
Si nosotros vamos ahora al KB Explorer, vemos que tenemos unas carpetas nuevas, donde podemos observar que hay distintos recursos del GAM que se fueron agregando en la medida de que este wizard fue importando objetos.



Por ejemplo, acá en Library están varias librerías que voy a poder utilizar del GAM, básicamente objetos externos de la API. También puedo obtener algunos ejemplos, y por supuesto, algunos objetos generales del GAM que se puede ver luego cómo funcionan.

Vamos a esperar unos minutos para que termine de importarse los objetos...

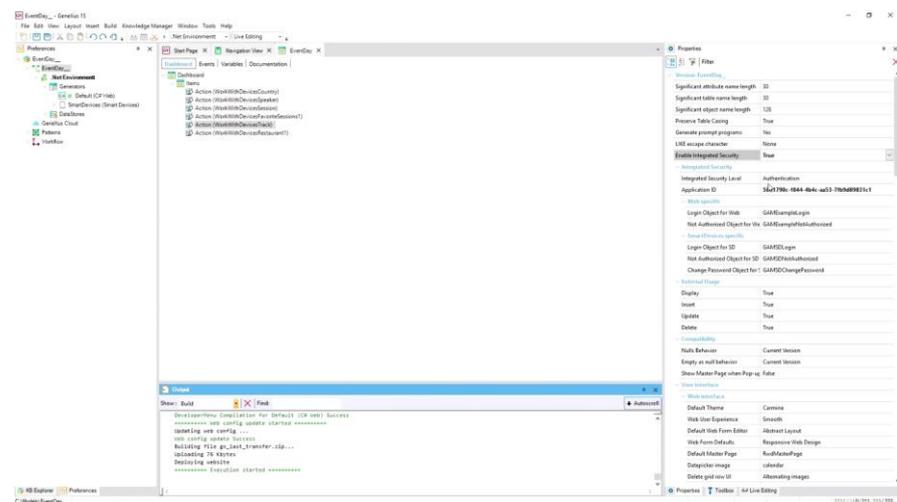
Muy bien. Ahí terminó de importarse los objetos y ahora lo que vamos a hacer es un Rebuild All. Así que vamos a la opción de Rebuild All, y ven que me pregunta que la base de datos del GAM no está encontrada, y me ofrece crearla.



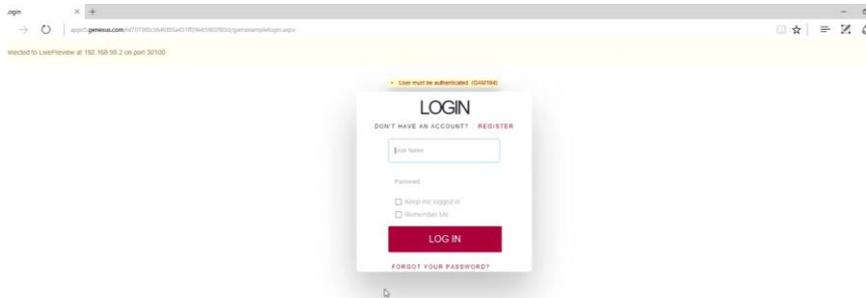
El GAM va crear su propia base de datos, donde se va a guardar allí toda la información relacionada a los usuarios, los permisos, etc. en una base de datos distinta a la base de datos de nuestra aplicación. Así que procedemos a decirle “Yes” para que se cree la base de datos.

Bueno, como vemos, terminamos de hacer el Rebuild All, así que ahora voy a ejecutar la aplicación...

Mientras tanto, podemos verificar, por ejemplo aquí en la propiedad Integrated Security Level, vemos que dice Authentication, que fue lo que nosotros marcamos, o sea que solamente vamos a ver la parte de autenticación; no vamos a ver autorización.



Bueno, ya se me abrió el Developer Menu, y vamos a buscar entonces ahora el objeto GAM Home. Ven que ahora para ejecutar la aplicación me pide un Login. Este Login se programó automáticamente; lo agregó el GAM con uno de los objetos que el GAM me agregó a mi KB. Lo único que tuve que hacer yo fue cambiar una propiedad y hacer un Rebuild All, nada más.



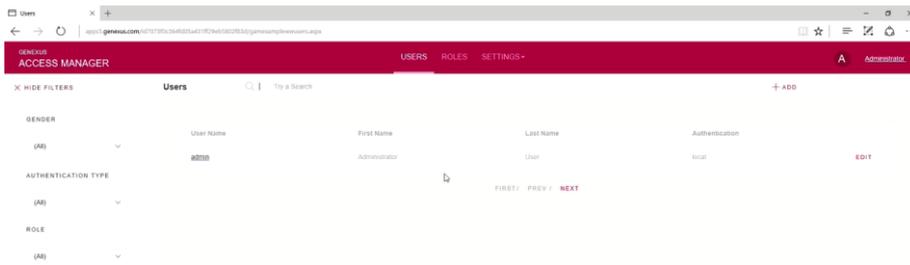
Entonces bueno, ahora me voy a loguear, el usuario es admin y la clave es admin123, por supuesto nosotros podemos cambiar esto, o podemos definirnos nuestros propios usuarios.

Di Login, y estoy ejecutando el objeto GAM home, que es el backend del GAM. Y ven que aquí tenemos varias opciones...

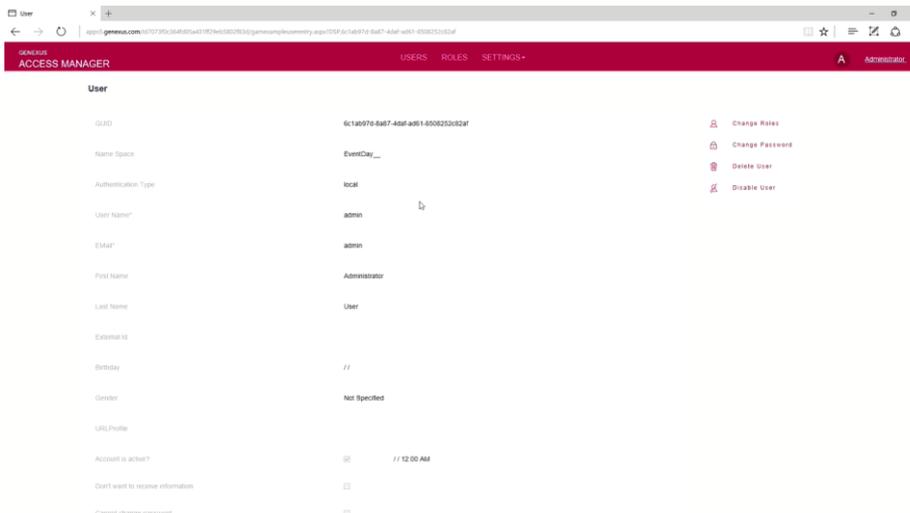


<http://test5.genexus.com:167279624462541152645829234/genexus/employees.aspx>

...por ejemplo voy a elegir user...



...y veo que el único usuario que tengo definido hasta ahora es el usuario administrador. Ven que acá arriba a la derecha también esa clave que yo ingresé, y ese usuario, corresponden al usuario administrador.



Aquí tengo varios datos referidos al administrador. Inclusive observen que puedo realizar distintas operaciones con el administrador. Pero en lugar de hacerlo con el administrador, vamos a crearnos un usuario nuestro, con autenticación local... el user va a ser training, y esto en principio lo podría dejar vacío: [training@genexus.com](mailto:training@genexus.com). Vamos a poner un password, "training" todo el minúscula, vamos a confirmarlo, y simplemente vamos a ponerle el primer nombre de training...

GENEXUS ACCESS MANAGER

USERS ROLES SETTINGS

User

Namespace: EventDay\_

Authentication Type: local

User Name: Training

Email: training@geneux.com

Password: \*\*\*\*\*

Password confirmation: \*\*\*\*\*

First Name: Training

Last Name:

External Id:

Birthday: //

Gender: Not Specified

Don't want to receive information:

Cannot change password:

...el resto lo vamos a dejar así, y vamos a ponerle seguridad por defecto, y confirm...

GENEXUS ACCESS MANAGER

USERS ROLES SETTINGS

User

Email: training@geneux.com

Password: \*\*\*\*\*

Password confirmation: \*\*\*\*\*

First Name: Training

Last Name:

External Id:

Birthday: //

Gender: Not Specified

Don't want to receive information:

Cannot change password:

Must change password:

Password never expires:

User is blocked:

Security policy: Default Security Policy

Enabled in applet:

CANCEL CONFIRM

Ahora vemos que tenemos un usuario también para training:

GENEXUS ACCESS MANAGER

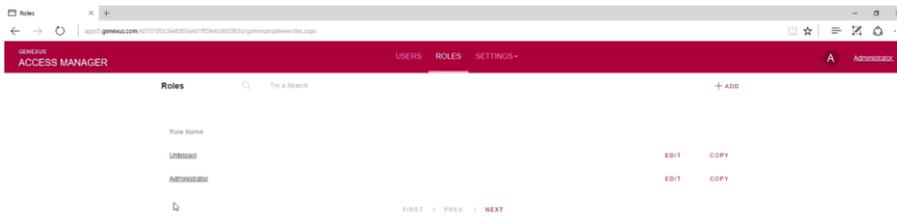
USERS ROLES SETTINGS

Users

USER NAME	FIRST NAME	LAST NAME	AUTHENTICATION	EDIT
admin	Administrator	User	local	EDIT
training	Training		local	EDIT

FIRST / PREV / NEXT

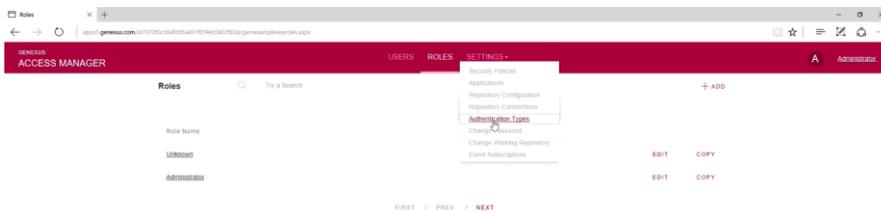
Aquí, en Roles, podemos definir los distintos roles, por ahora tengo solamente definido un rol de administrador...



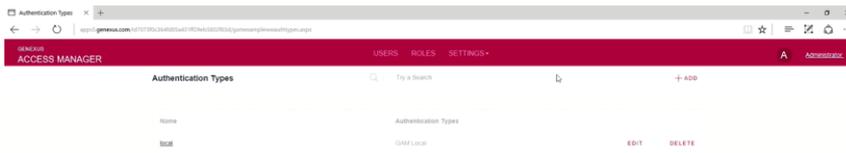
...pero podríamos crear otros roles dependiendo de lo que yo tenga en mi empresa, por ejemplo un DBA, o un rol de gerente, o distintos roles para los cuales voy a definir después distintos permisos que puedan entrar a algunas partes de la aplicación y a otras no.

Por ahora, estamos probando solamente autenticación, así que lo que estamos probando es que me pida un usuario, una clave, y poder verificar que la persona que esté ingresando es la persona correcta.

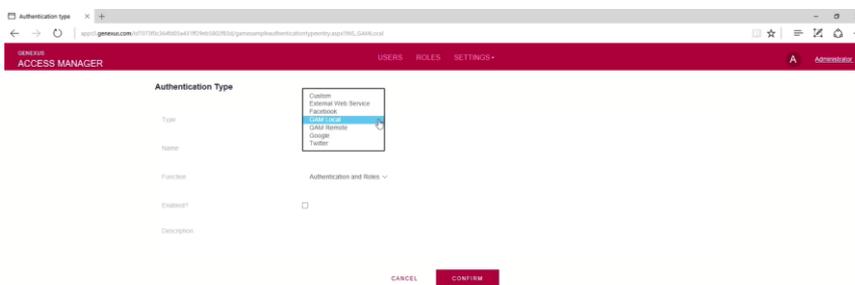
Ven que aquí en Settings tengo también otras opciones...



...aquí por ejemplo tengo solamente creada la autenticación local...



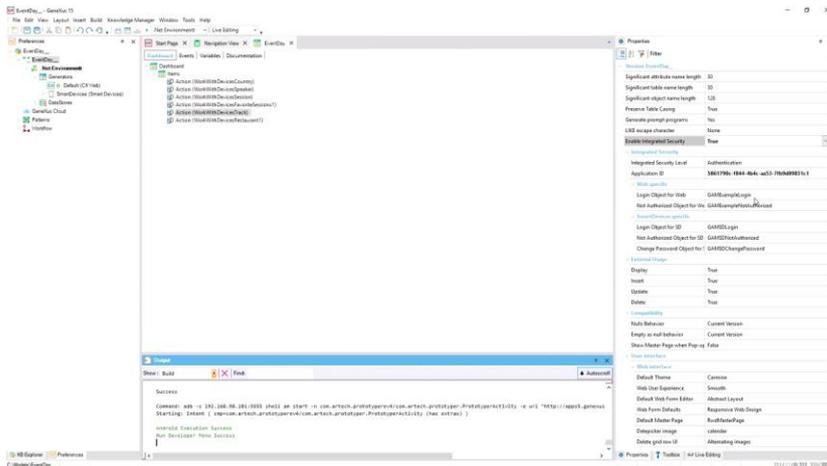
... podría agregar otro tipo de autenticación...



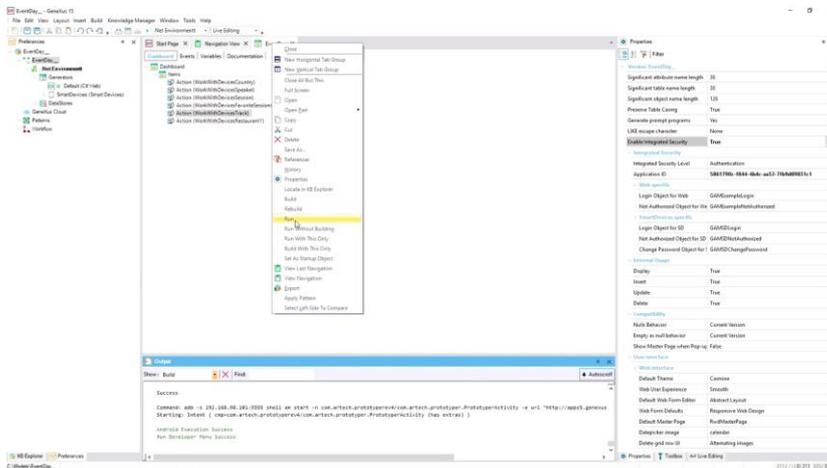
Recuerden que vimos que podían ser autenticaciones tipo Custom, o Facebook, o Google, o Twitter, es decir tengo distintas maneras de poder autenticarme, y aquí yo tengo que definir para mi aplicación cuál es el tipo de autenticación que yo voy a desear.

Bien. Vamos a volver a GeneXus, y quería mostrarles que ese objeto Login que apareció lo podemos ver acá, es el GAMExampleLogin, que automáticamente creó el GAM, y que me quedó por defecto asociado, pero yo podría crearme mi propia pantalla de Login y ponerla en mi aplicación.

También tengo una pantalla en el caso de que no se autorice el Login, y también para SD. En el caso de Smart Devices yo tengo la posibilidad de definir también un objeto de Login, en particular ahora está usando el GAMSDLLogin.

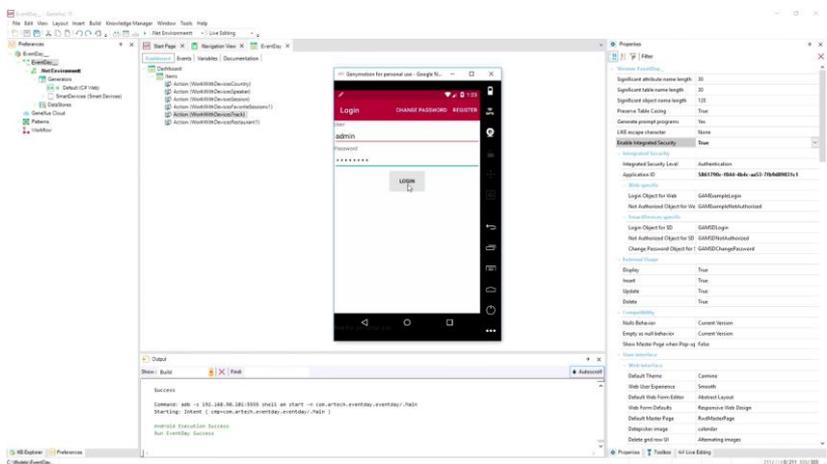


Voy a ejecutar ahora el objeto para Smart Devices, en este caso el objeto main es el EventDay, que vendría a ser el menú...

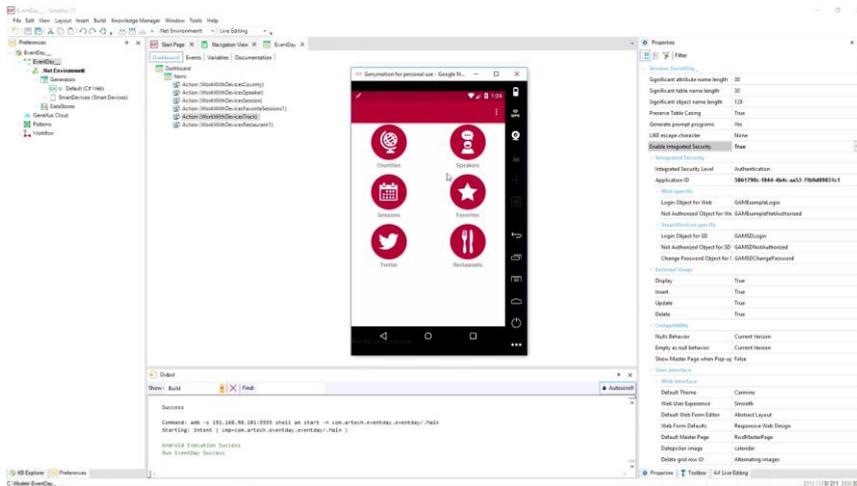


...y vamos a observar que también se va a agregar seguridad a la aplicación para Smart Devices, o sea que no va a poder entrar directamente al menú, sino que me va a tener que pedir que me loguee primero.

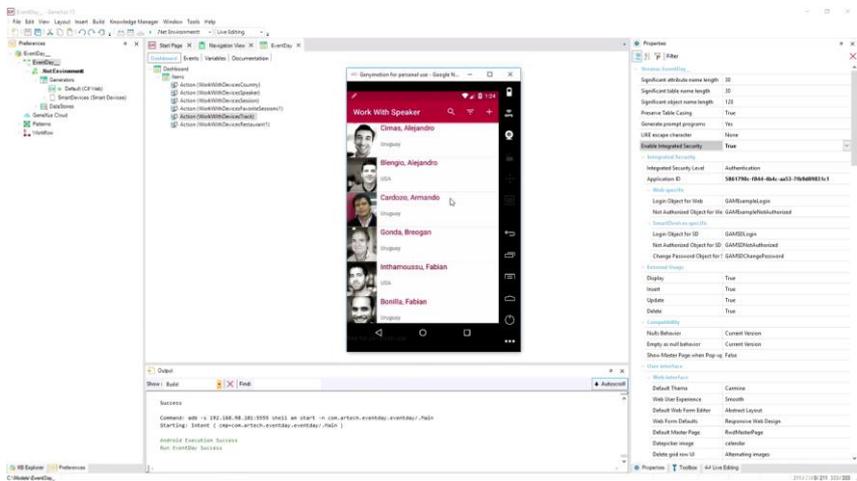
Ven, aquí tengo el emulador, y me abrió una pantalla de Login para Smart Devices. Así que voy a ingresar, como hice hoy... admin, admin123...



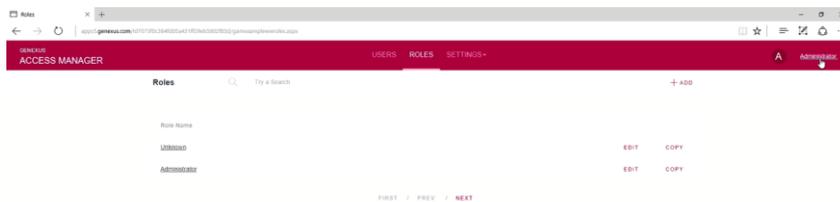
...y veo que ahora sí efectivamente tengo acceso al menú de la aplicación...



... y por supuesto puedo ver en este caso los Speakers, etc.



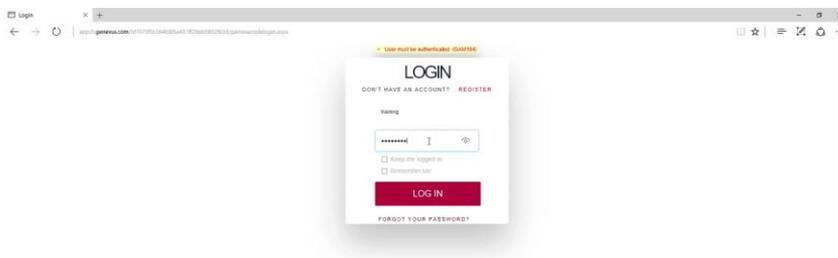
Bueno, vamos a volver a la pantalla web, estábamos aquí en Roles...



...ven que yo estaba logueado como administrador; lo que voy a hacer ahora es un Logout...



...y me voy a logear con el usuario que había creado antes: el usuario de training, con password "training".

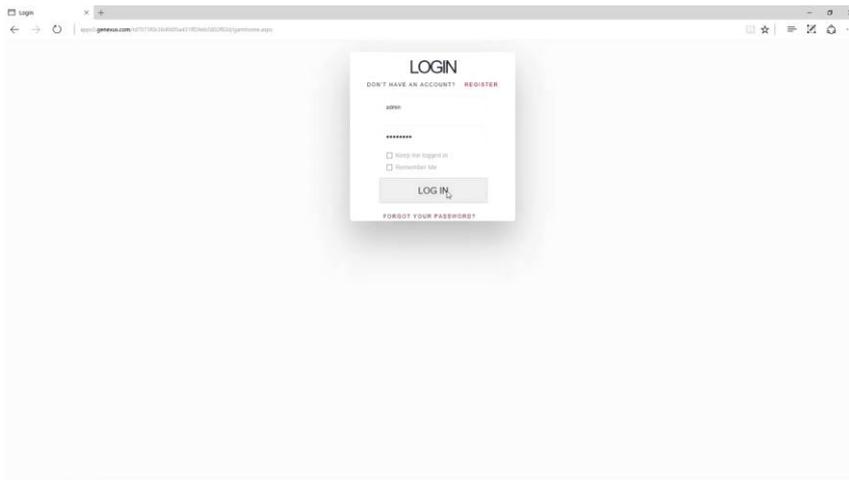


Me dice que este usuario no está autorizado; esta es otra de las pantallas automáticas que les decía.

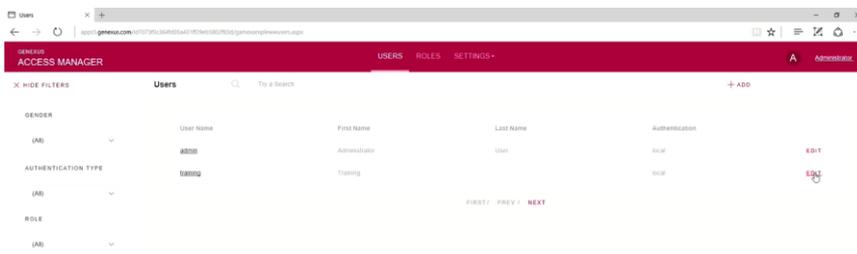


Entonces lo que vamos a hacer es asociarle un rol al usuario, por ejemplo el rol de administrador.

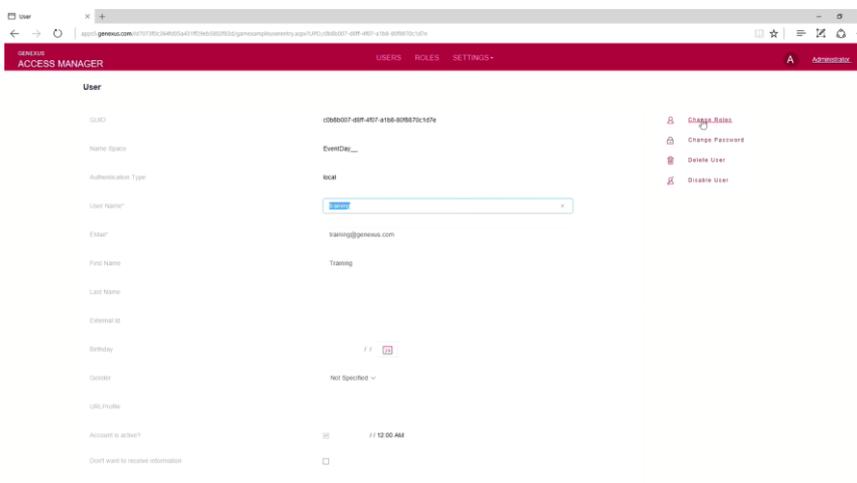
Así que primero, voy a logearme otra vez como administrador.



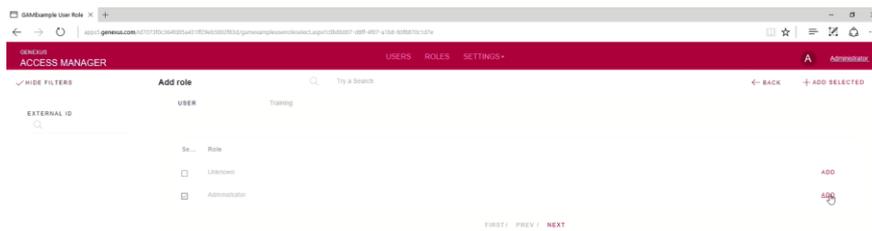
Así que lo que voy a hacer ahora es asignarle al usuario training el rol de administrador. Vamos a editar el usuario....



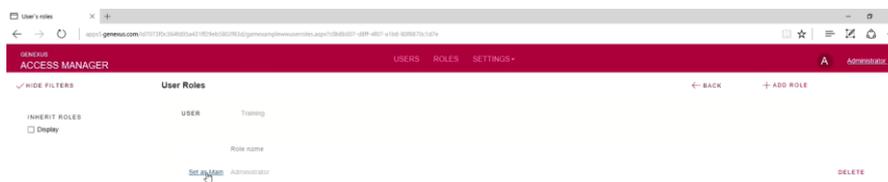
...vamos a cambiarle los roles...



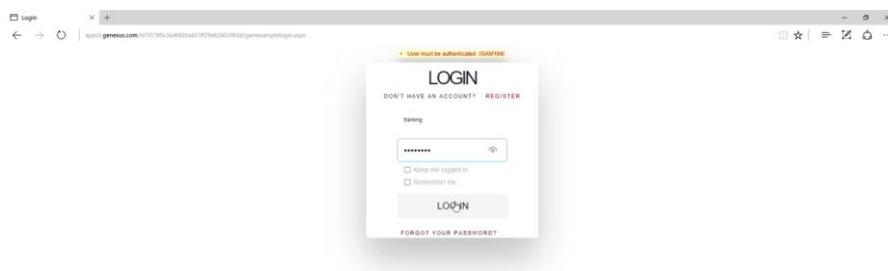
...y voy a agregar el rol de administrador.



Ahora voy a setear que este rol sea el rol principal.



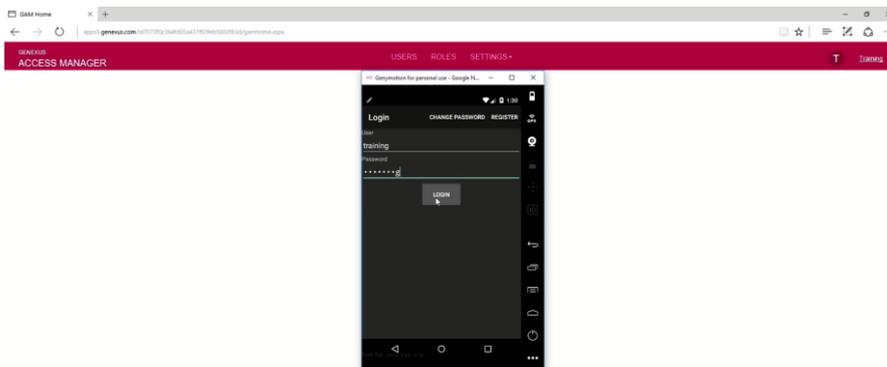
Muy bien. Ahora voy a probar otra vez desolguarme y loguearme con el usuario de training...



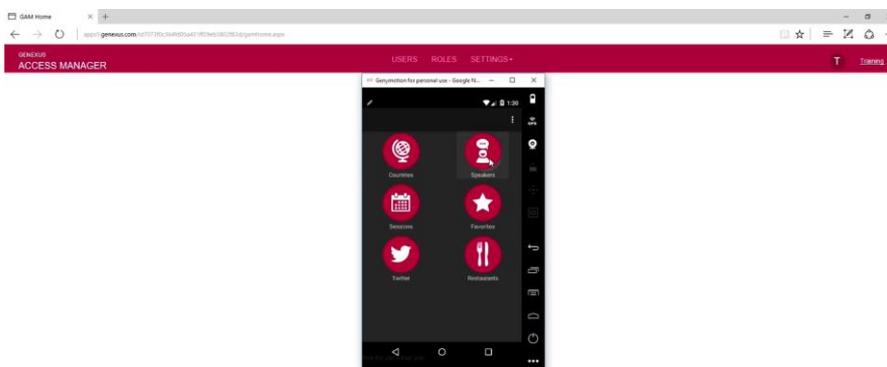
...y vemos que me pude loguear ahora con el usuario de training al backend del GAM.

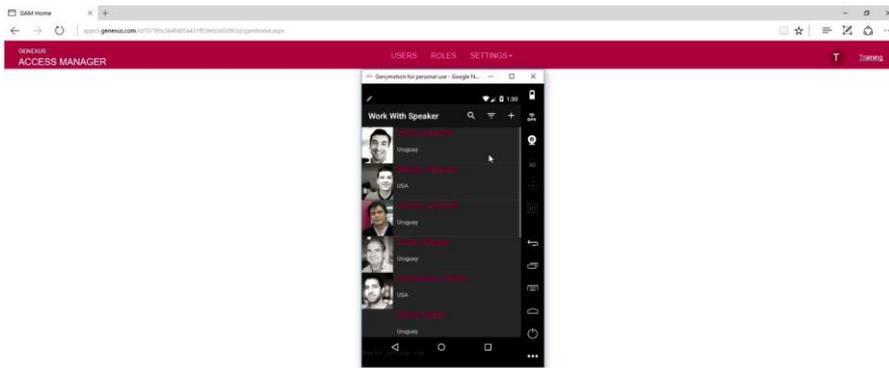


Vamos a ir entonces al emulador. Me pide loguearme, así que voy a usar el mismo usuario...



...y ven que el usuario de training tiene acceso entonces a los datos en la pantalla.





---

Bueno, esta fue una pequeña demo para mostrar qué tan fácil es agregarle seguridad a la aplicación. Es cambiar una propiedad, hacer un Rebuild All, y después ya tenemos un backend que nos permite definir usuarios, permisos, prácticamente sin hacer nada; inclusive hasta los objetos los tenemos prontos, como para poder hacer el Login in validar que las personas que tienen que entrar sean realmente las que nosotros queremos que ingresen al sistema.

En este video vimos los beneficios del GeneXus Access Manager para proveernos una solución completa de seguridad para nuestra aplicación, de forma muy sencilla.

Hay mucha más información sobre este tema. Para saber más, los invitamos a visitar el link que se ve en pantalla.



More info:

<http://wiki.genexus.com/commwiki/servlet/wiki?24746>