

## Uso de la API del GAM



En videos anteriores, vimos como construir una aplicación para una inmobiliaria, con una parte Web y una parte para Smart Devices

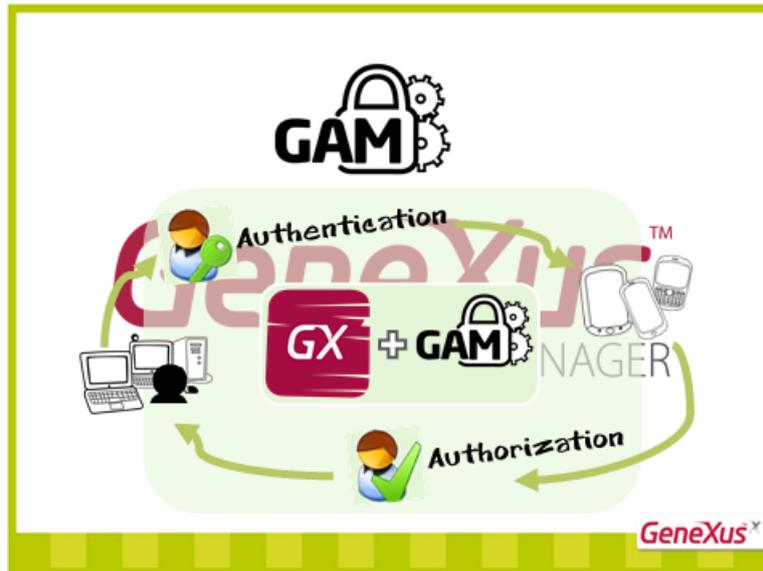
Y luego vimos como resolver la Seguridad de ambas aplicaciones, utilizando para esto el GAM, GeneXus Access Manager, configurando la propiedad a nivel de la Versión, Enable Integrated Security con el valor True.



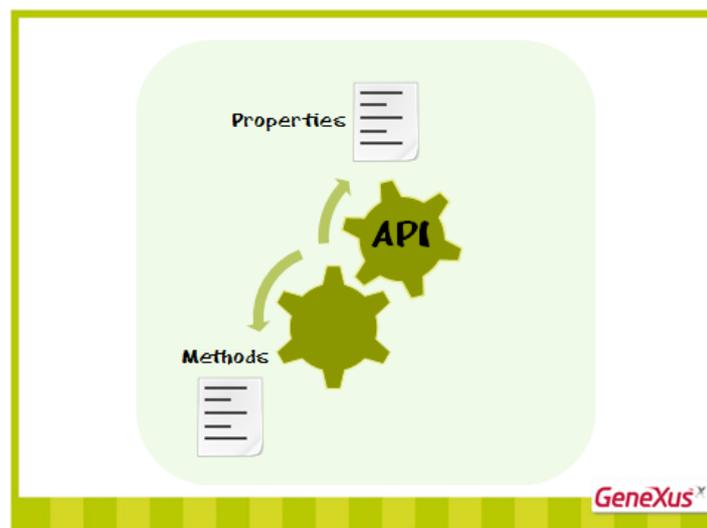
El GAM es un módulo de Seguridad desarrollado en GX, que resuelve los problemas de autenticación y autorización, tanto para aplicaciones Web como para aplicaciones para Smart Devices.

Lo que se pretende con el GAM es que la solución de Seguridad se utilice de la forma más declarativamente posible.

Y esto se logra fácilmente, ya que el GAM es un Módulo de Seguridad desarrollado en una **KB GX**, que se integra a nuestra aplicación y permite resolver de manera centralizada todo lo referente a la Seguridad de la misma.



Además, el GAM expone una API para acceder a diferentes métodos y propiedades en caso de que sea necesario hacerlo desde nuestra aplicación a la KB del GAM.

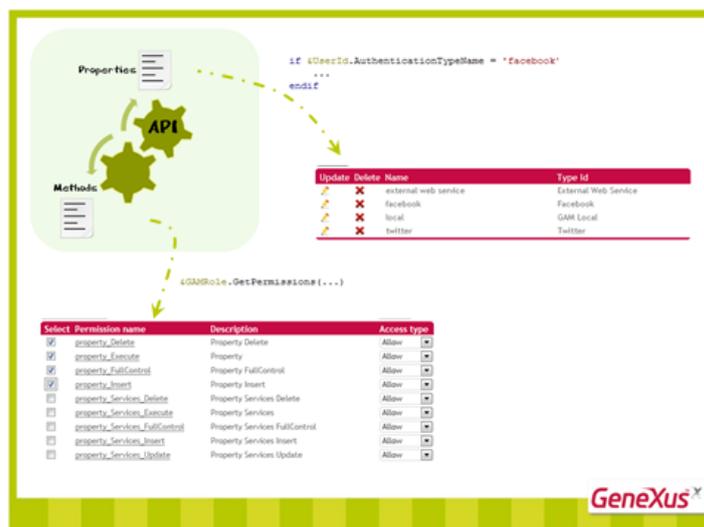


El acceso a las propiedades y métodos de la API se realiza a través de external objects que se importan al habilitar el GAM en nuestra aplicación, y podemos realizar diferentes acciones, como por ejemplo

- Obtener el usuario que está logueado en caso que necesitemos utilizarlo desde dentro de nuestra aplicación, o en lugar del nombre, tal vez sea necesario acceder a su dirección o cualquier otra información relacionada con el usuario



- También podríamos necesitar los tipos de autenticación configurados en nuestra aplicación
- O tal vez podríamos necesitar los permisos asociados a un rol, para realizar alguna acción con ellos dentro de nuestra aplicación

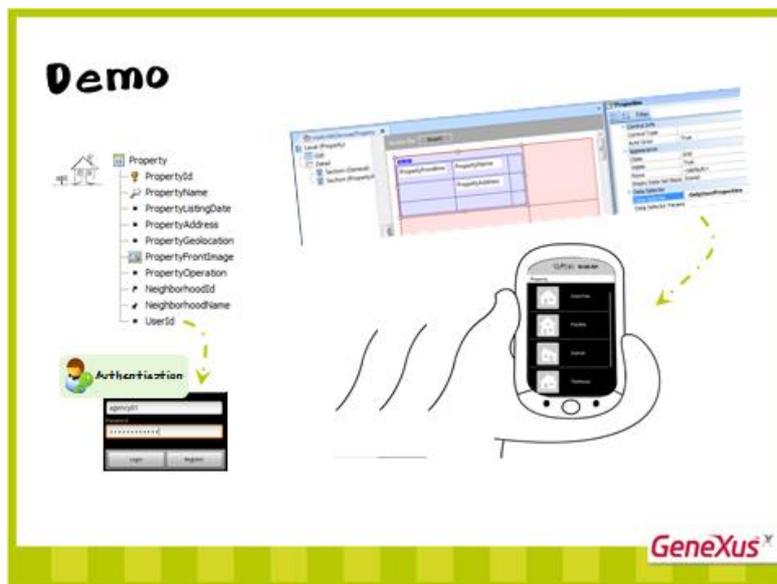


Veamos un ejemplo de uso de la API.

Como hemos visto, los agentes inmobiliarios de nuestra aplicación pueden ingresar y visualizar las propiedades que poseen en venta o alquiler accediendo a la aplicación para Smart Devices, desde sus dispositivos.

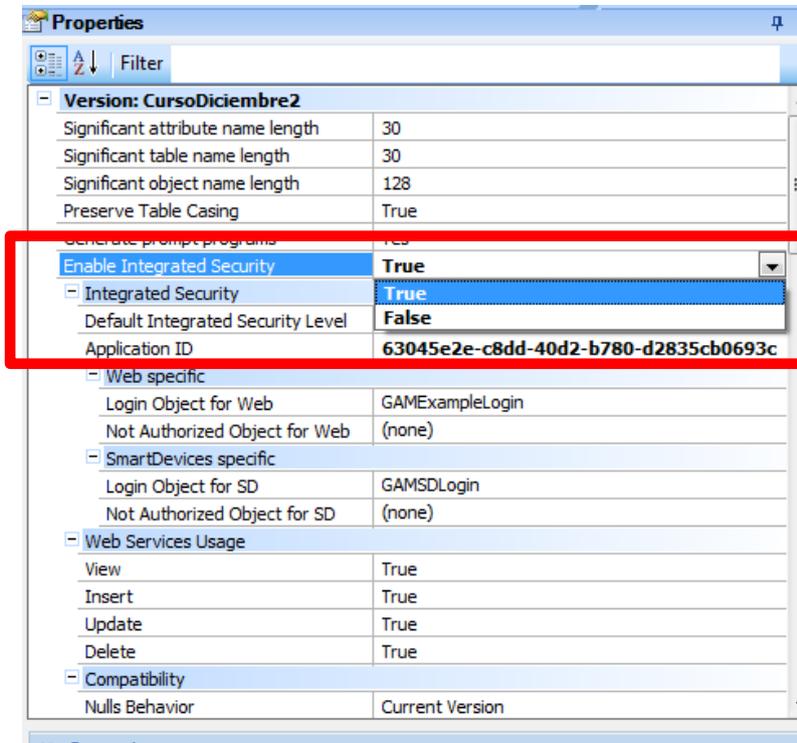
Pero como la Transacción Property es única para todos los agentes inmobiliarios, debemos identificar al agente que ingresa una propiedad en el sistema. Es por esto que debemos almacenar en la Transacción Property, la identificación del agente inmobiliario que ingresa la propiedad.

Además, en el List de Propiedades que se muestra en la aplicación para Smart Devices, se requiere que los agentes inmobiliarios solo puedan visualizar los registros ingresados por ellos y no la información ingresada por otro agente.



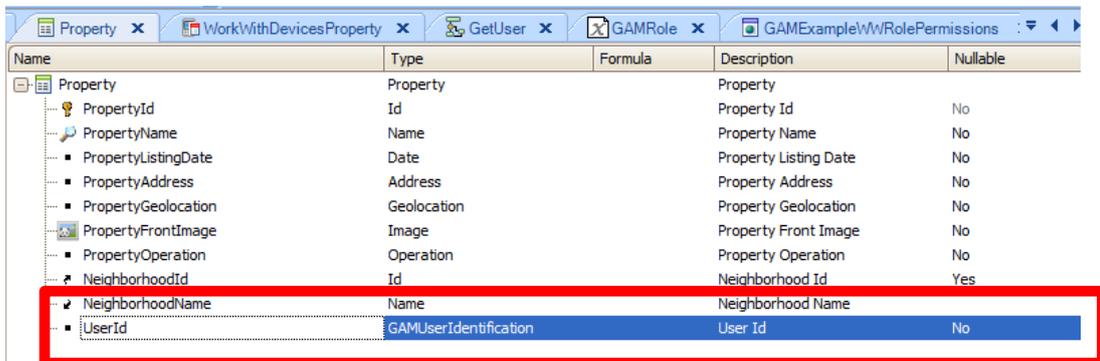
Ya tenemos creada nuestra aplicación Web y Smart Devices para la inmobiliaria.

Como podemos ver nuestra aplicación tiene el GAM aplicado.



Vamos ahora a la Transacción Property a definir un nuevo atributo que nos va a permitir almacenar al usuario que se intenta loguear a través del GAM.

Esta atributo se va a llamar UserId, y será del tipo GAMUserIdentification.



Name	Type	Formula	Description	Nullable
Property	Property		Property	
PropertyId	Id		Property Id	No
PropertyName	Name		Property Name	No
PropertyListingDate	Date		Property Listing Date	No
PropertyAddress	Address		Property Address	No
PropertyGeolocation	Geolocation		Property Geolocation	No
PropertyFrontImage	Image		Property Front Image	No
PropertyOperation	Operation		Property Operation	No
NeighborhoodId	Id		Neighborhood Id	Yes
NeighborhoodName	Name		Neighborhood Name	
UserId	GAMUserIdentification		User Id	No

Veamos que este tipo GAMUserIdentification es un dominio que se importa junto con los objetos del GAM.

En las reglas de la Transacción declaramos

```
Equal(UserId, GetUser());
```

De esta forma lo que estamos haciendo es capturando la información del usuario que se está logueando, y esta información se va a guardar en la Transacción Property al momento de ingresar una propiedad.

En cuanto a GetUser, se trata de un procedimiento que tiene declarada la siguiente regla Parm, y cuya lógica nos permite capturar al usuario que está logueado en nuestra aplicación.

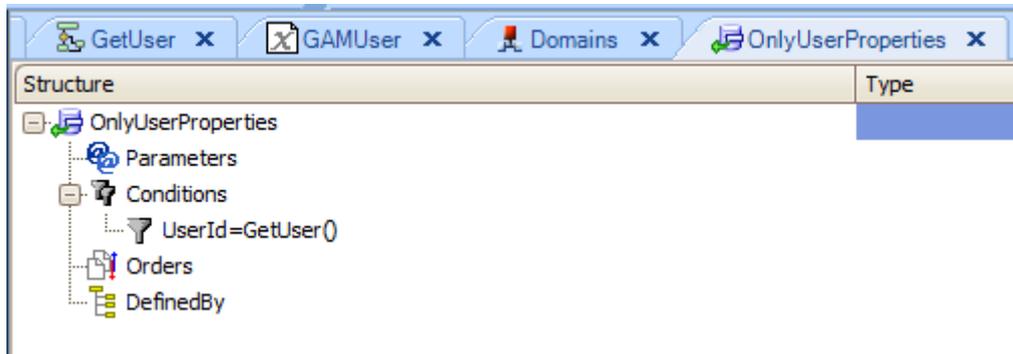
GAMUser es una variable estática que instancia la información del usuario logueado, pudiéndose utilizar métodos asociados como GetName, GetEmail, etc. para obtener la información relacionada.

En este caso utilizamos el método GetName para obtener el nombre del usuario logueado.

De esta forma al ingresar una nueva Propiedad a la aplicación, se obtendrá la información del agente inmobiliario logueado y se salvará esta información en la tabla al momento de confirmar el registro.

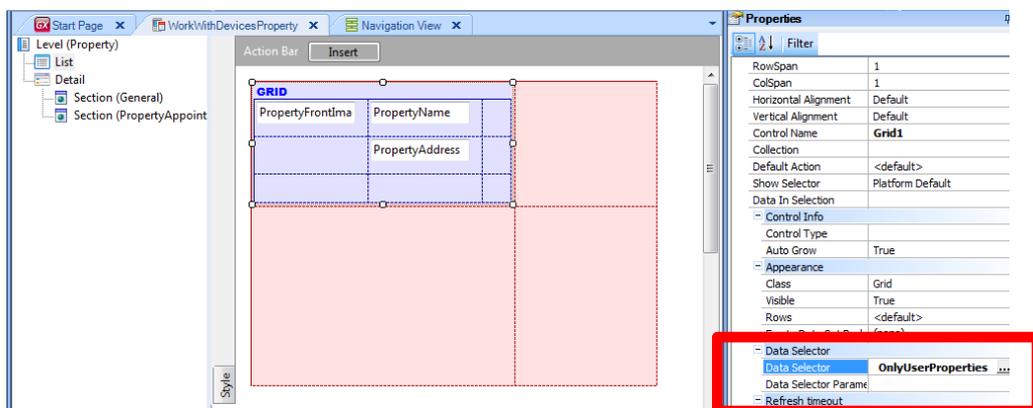
Veamos ahora qué debemos hacer para que desde el List de Propiedades de la aplicación para Smart Devices, cada agente inmobiliario vea solamente sus propiedades.

Para esto hemos creado el Data Selector que estamos viendo.



Observemos que el procedimiento GetUser es el que hemos creado recientemente.

Vamos ahora al Work With Devices Properties, seleccionamos el List y en el grid vamos a la propiedad Data Selector, y le asociamos el Data Selector nuestro creado recientemente.



Ya están listos los cambios en GeneXus, así que presionamos F5.

Primero se ejecuta la reorganización para agregar el nuevo atributo UserId a la transacción Property.

Vamos a loguearnos con el usuario agency01 para registrar una nueva propiedad.

Observemos primero que ya existen dos propiedades registradas. Una de ellas a nombre del usuario admin, y otra a nombre del usuario agency01.

Vamos entonces a registrar una nueva propiedad a nombre del usuario agency01.

El nombre de la propiedad es Utopía, ponemos la fecha, la dirección, vamos a ubicar la imagen, el barrio donde se encuentra que es Coral Gable, y vamos a ver que al presionar el botón Confirm aparece efectivamente el usuario agency01 como el responsable del ingreso de esta propiedad.

Prop[Next]ty

Id: 6

Name: Magnolia

Listing Date: 02/16/12

Address: 123 Sweet Street

Geolocation: [Empty field]

Front Image: [Image of a house]

Operation: Sale

Neighborhood Id: 2

Neighborhood Name: Coral Way

Id: agency01

Buttons: Confirm, Cancel, Delete

Vamos ahora a la aplicación para Smart Devices, nos vamos a loguear con el usuario agency01, para ver que efectivamente las propiedades que visualizamos son las ingresadas por este agente inmobiliario.

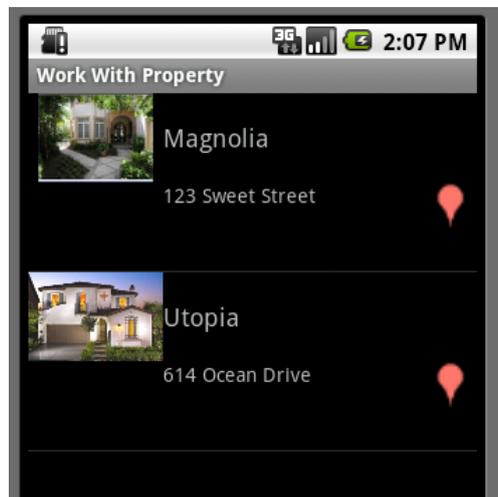
User: agency01

Password: [Masked]

Buttons: Login, Register

Time: 2:08 PM

Vamos al List of Properties, y vemos efectivamente las dos propiedades ingresadas por este agente.



De esta forma, el haber habilitado el GeneXus Access Manager en nuestra aplicación nos ha permitido hacer uso de los métodos y propiedades de la API fácilmente, y así obtener información sobre el GAM y poder utilizarla desde nuestra aplicación.

Es decir, la API nos ha permitido extender la funcionalidad de los mecanismos de seguridad provistos por el GAM.

