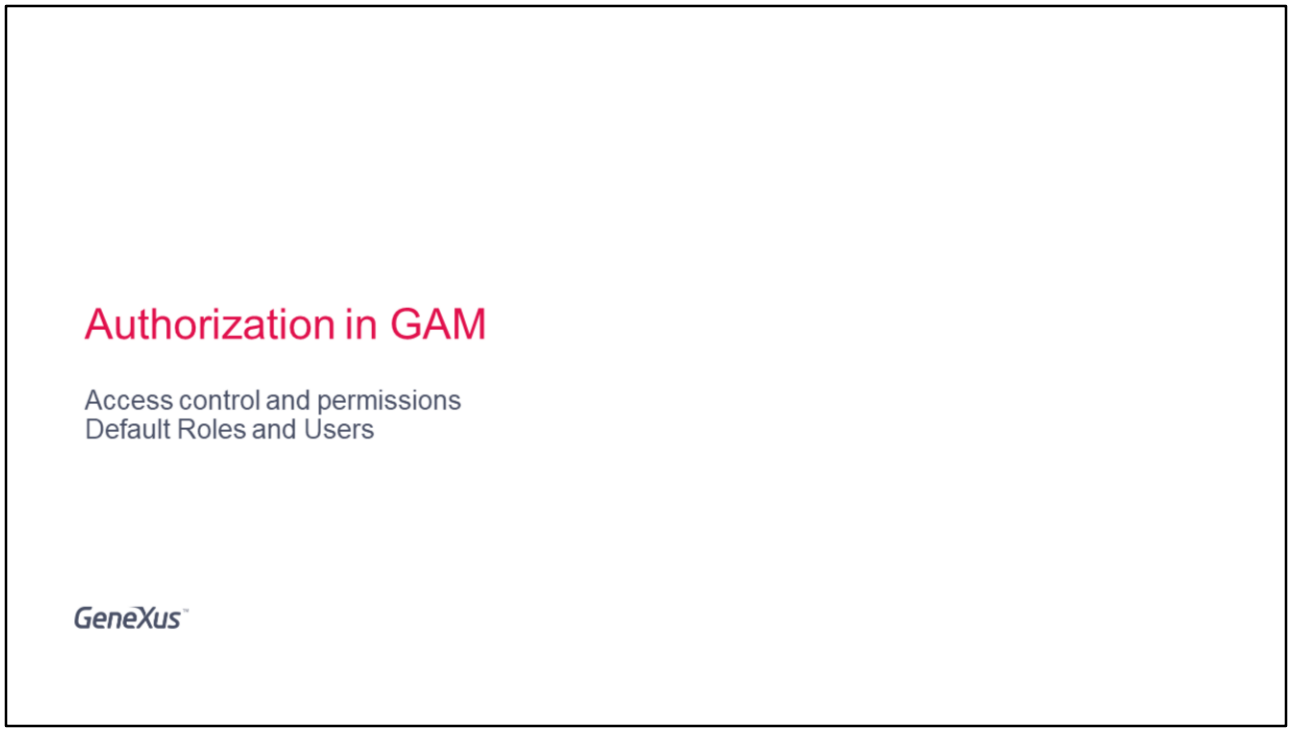

GeneXus™
by Globant

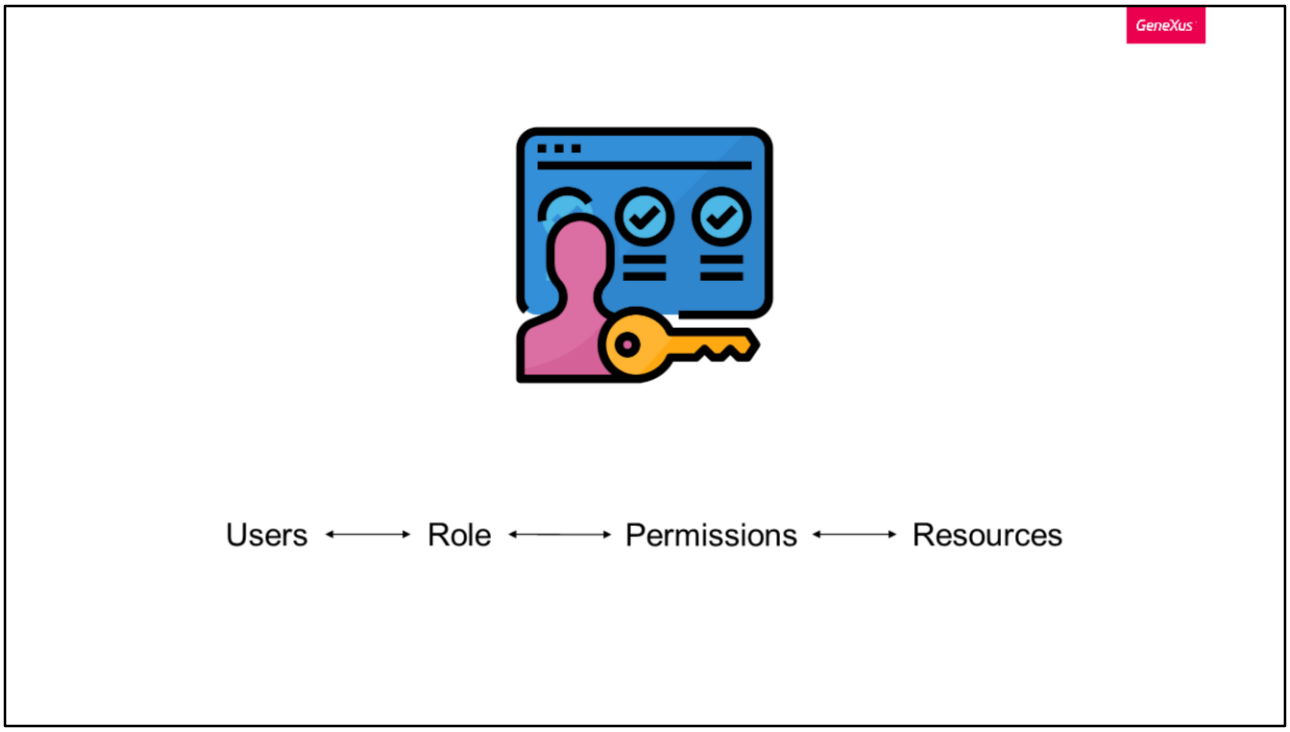
Authorization

Nicolas Adrién

GeneXus



En este video trataremos los temas relacionados a la Autorización en GAM.
Control de acceso y permisos, así como también roles y usuarios por defecto.

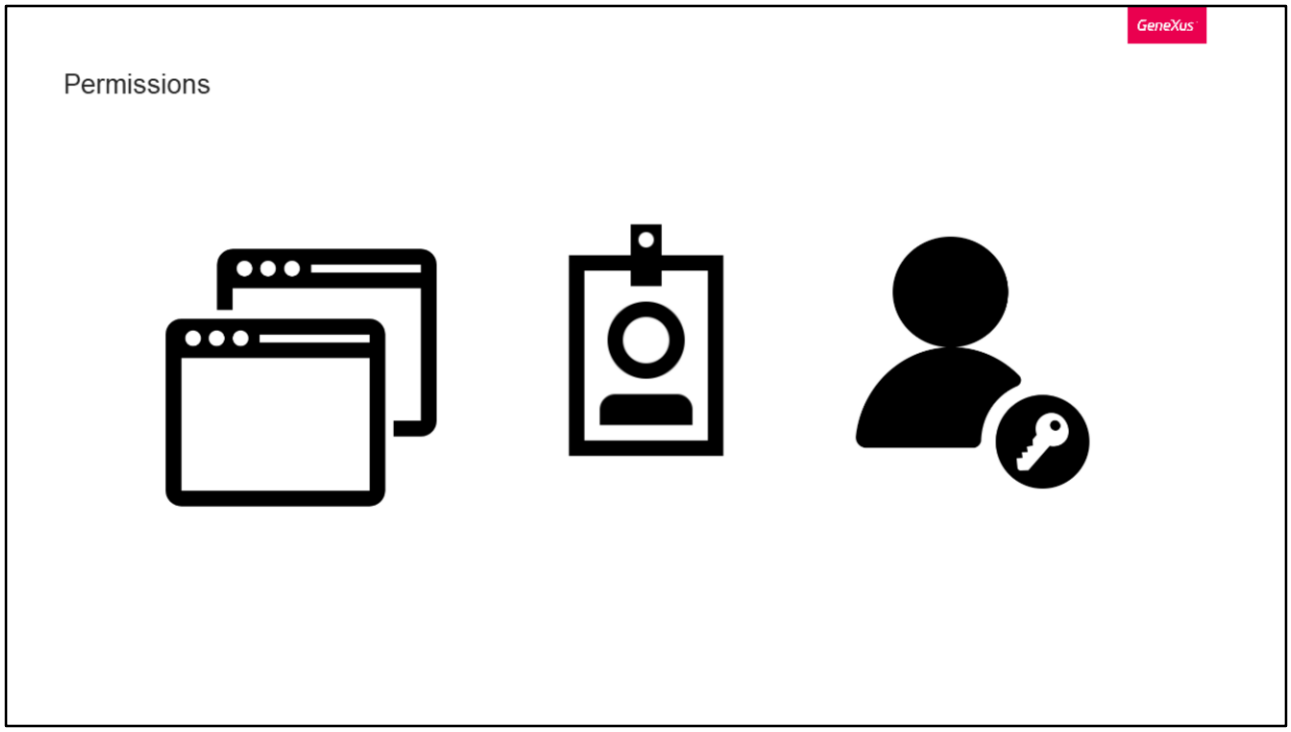


Además de la Autenticación, como se menciona en el curso introductorio, tenemos el concepto de Autorización.
La autorización es el proceso de verificar si un usuario que ya fue autenticado, posee los permisos necesarios para realizar una o más acciones en el sistema.

Para esto, como comentamos en videos anteriores GAM cuenta con un esquema basado en Roles de Usuario, donde cada usuario tiene asociado uno o varios Roles. También se tienen los Recursos asegurados y la asignación de Permisos sobre estos Recursos a los Roles.

Los recursos pueden ser, por ejemplo:

- Web Panels o paneles Móviles
- Work With para dispositivos móviles.
- Web Components con Acceso por URL habilitado
- Transacciones WEB
- Entre otros

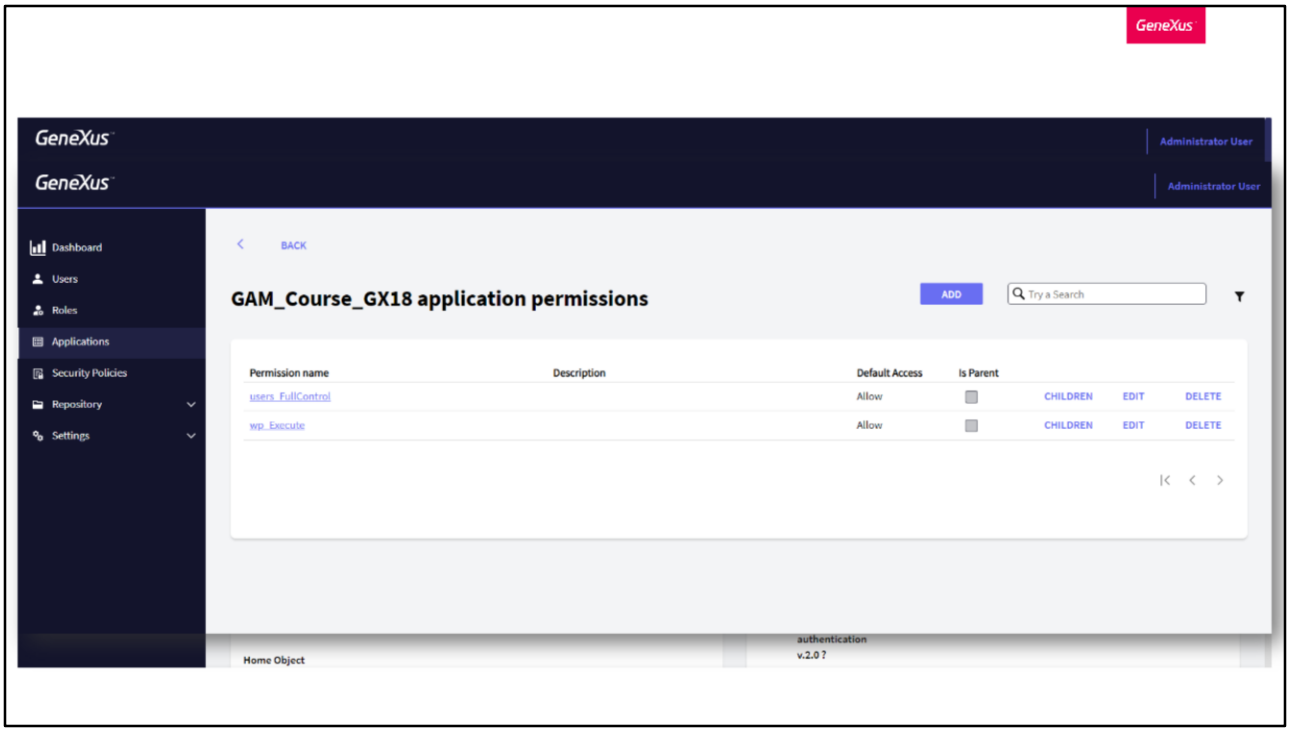


Los permisos utilizados para la autorización, se pueden asignar en diferentes niveles:

Nivel de aplicación: Cada uno de los Permisos tiene un Tipo de Acceso definido a nivel de Aplicación GAM, donde se define un Tipo de Acceso Predeterminado para cada permiso.

Nivel de rol: Cuando se asignan Permisos a Roles, se definen con un Tipo de Acceso.

Y finalmente, Nivel de usuario, que es cuando los Permisos se asignan a los Usuarios, y se definen con un Tipo de acceso al igual que a nivel de rol.

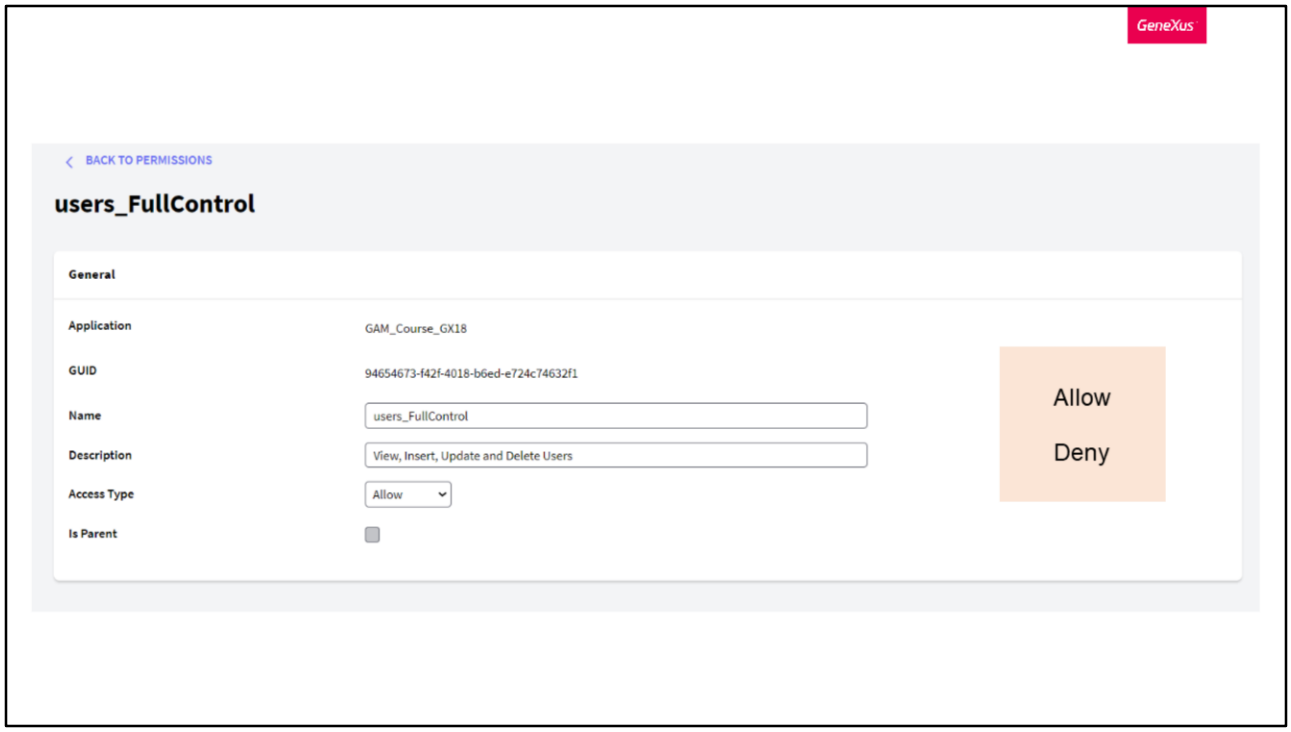


A nivel de Aplicación, GAM brinda la facilidad de que a cada aplicación generará automáticamente los permisos.

Para acceder a dichos permisos, basta con dirigirnos a la Aplicación en cuestión desde el Backend, y presionar en Permisos, dentro del sub menú “Más opciones”.

Una vez allí, encontraremos todos los permisos que se encargó de generar GeneXus por nosotros, los cuales se conforman de un nombre, descripción, acceso por defecto y si es padre o no de otro permiso.

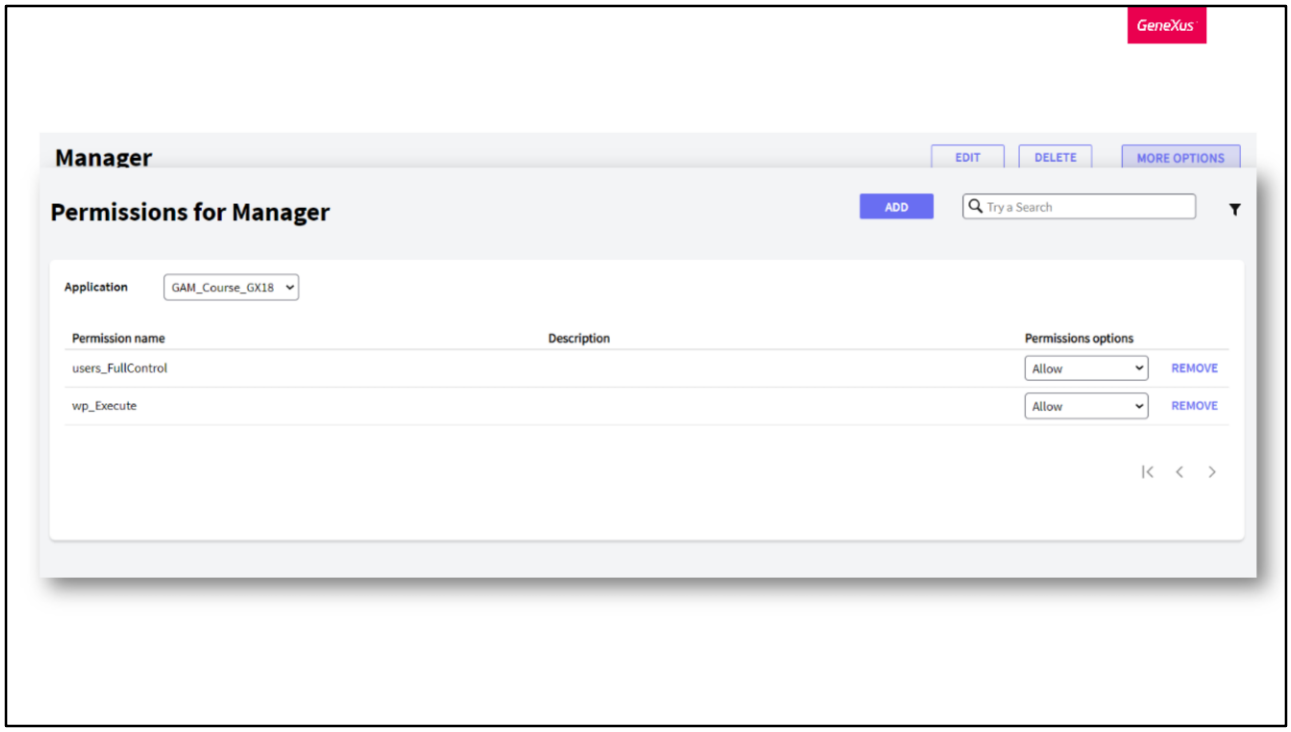
Para este ejemplo que vemos en pantalla, tenemos un permiso de ejecución (por eso termina en Execute) y otro de control total, pero también se pueden generar para los modos de Insert, Update y Delete. El permiso de control total representa la obtención de los permisos mencionados anteriormente.



En caso de querer editar un permiso, las opciones disponibles son poder cambiar el nombre, la descripción, y el tipo de acceso.

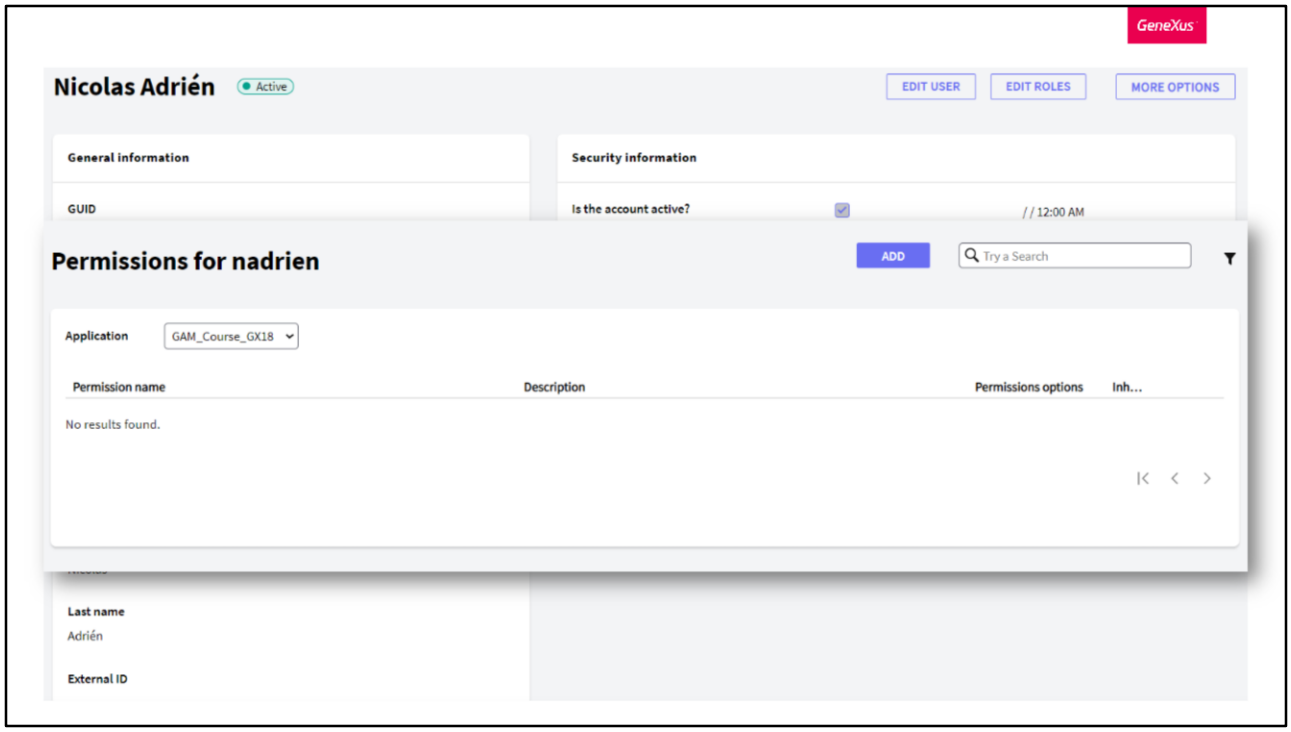
El tipo de acceso de un permiso define su uso predeterminado (si es algo público o restringido). Las opciones disponibles son:

- Permitir: Este tipo de acceso habilita el permiso para todos los usuarios por defecto. Los usuarios que tengan este permiso otorgado con: Tipo de Acceso = restringido o denegado, o que tengan algún rol en el que el permiso esté restringido o denegado, no tendrán este permiso.
- Restringido: Los usuarios no tienen este permiso por defecto, lo cual implica que sólo los usuarios que tienen este permiso otorgado con Tipo de Acceso = Permitir o tienen algún rol donde se permite este permiso, tienen los derechos correspondientes.



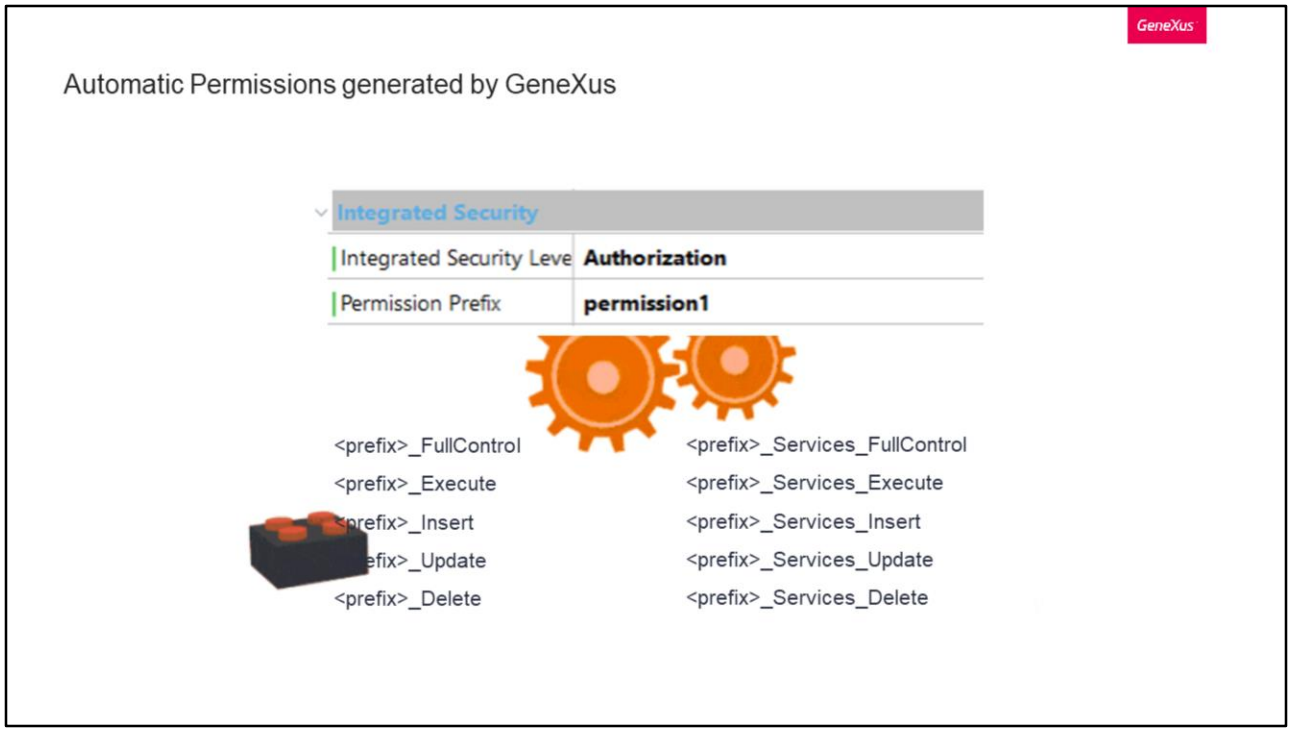
A nivel de Rol, para agregar, editar o borrar permisos debemos dirigirnos a la opción Roles y dentro de un rol podemos manejar sus permisos a través del sub menú “Mas opciones”.
Como vemos en la imagen, GAM nos brinda la posibilidad de Agregar o Eliminar un permiso, modificar su nivel como dijimos antes (con las opciones Permitir, Restringido y Denegado), y marcarlo como heredado o no.

Algo a destacar de esto, es que si agregamos un permiso a un rol, por transitividad éste permiso también lo tendrán todos los usuarios que tengan ese rol.



Finalmente, el ultimo nivel es por Usuarios.
Al igual que por Roles, primero se accede al Usuario y luego al editarlos, tenemos la opción de editar permisos dentro del sub menú “Mas opciones”.

Acá se manejan exactamente igual que por Roles como fue mencionado recién.
Un detalle a destacar es que los permisos otorgados en este nivel le ganan a los permisos otorgados a nivel de rol, sin importar el tipo de acceso que se tenga, el nivel de Usuario siempre va a ganar.



Muchos de los permisos que se veían en las capturas anteriores, correspondían a permisos automáticos autogenerados por GeneXus.

Al realizar Build, estos permisos son generados y luego se chequean en tiempo de ejecución.

Esto suponiendo que se tiene la propiedad Nivel de seguridad integrada establecida en el valor de Autorización por supuesto.

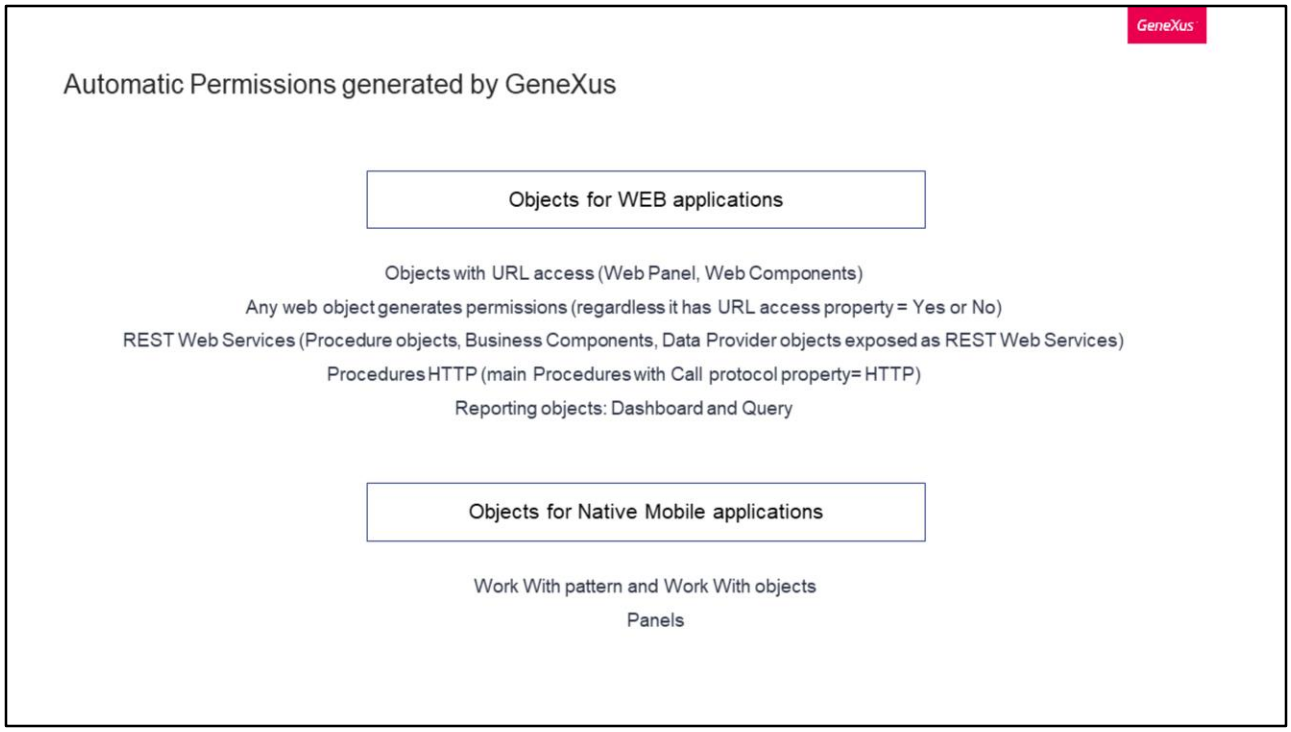
El código para verificar estos permisos está incluido en el código generado, y el usuario solo declara (a través de la propiedad Permission Prefix) cuál es el permiso que se va a verificar. Algo positivo a destacar de esto es que como se ve, no se necesita programar nada, solo basta con declarar los permisos necesarios para ejecutar el objeto.

Los permisos automáticos se pueden describir de la siguiente manera:
En primer lugar tenemos a los Permisos de ejecución donde cada objeto de la KB (excepto el Menú) expone un permiso de acceso. Mas adelante entraremos en detalle de cuales son los objetos que exponen los permisos.

En segundo lugar tenemos a los Permisos para la ejecución de las diferentes modalidades de una transacción.

Cuando se especifica un prefijo de permiso en cualquier transacción web (supongamos que es "prefix"), se crea un conjunto de permisos en el Repositorio GAM, llamados de la siguiente manera:
prefix.FullControl es el padre del resto de los permisos, y la representación de cada permiso está dada por la acción que se le concatena al prefijo.

En tercer lugar tenemos los Permisos de Servicios.
Si "prefix" es el prefijo de permiso de un componente comercial expuesto como REST, los siguientes permisos se generan automáticamente.



En la placa anterior, decíamos que cada objeto de la KB (excepto el Menú) expone un permiso de acceso. Veamos cuales son estos objetos.

Para aplicaciones web tenemos:

- Objetos web con acceso por URL como lo son los Web Panel y Web Components
- A partir de GeneXus Evolution 3, cualquier objeto web genera permisos, independientemente de que tenga propiedad de acceso URL con valor Si o No
- Servicios web REST, como lo son procedimientos, Business Components o Data Providers de datos expuestos como servicios web REST
- Procedimientos HTTP, los cuales son Main con propiedad de protocolo de llamada con valor HTTP

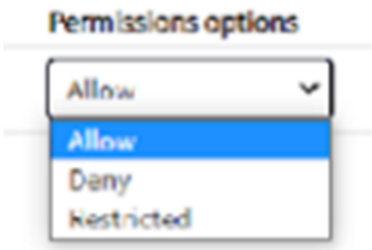
Y finalmente,

- Objetos de informes, como son los Dashboard y Querys

Para aplicaciones móviles nativas tenemos Work With pattern y Work With objects, y tambien a los Paneles.

GeneXus

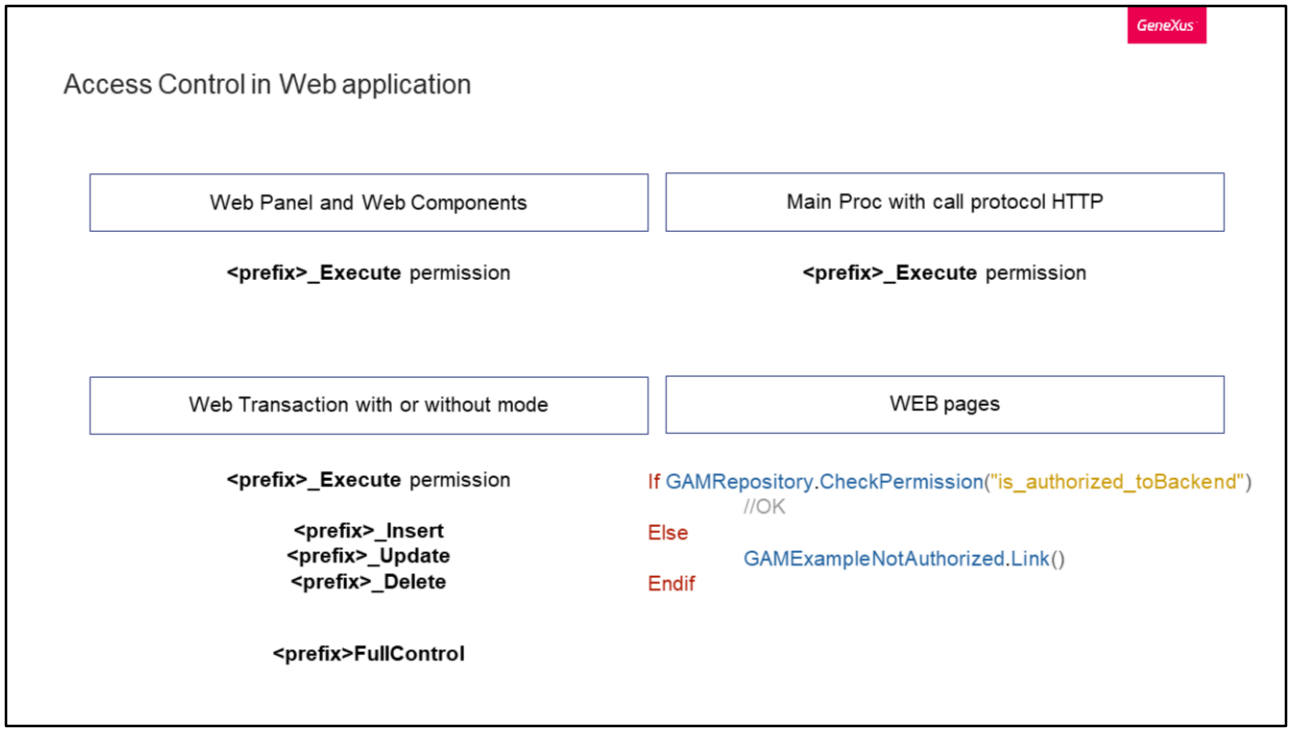
Permission denial



GAM permite especificar negación de permisos. Esto es, indicar que un permiso no puede ser utilizado en una sesión cuyo usuario posea dicho rol.
Ya vimos la opción anteriormente donde teníamos además de las opciones de Permitir y Restringir, Denegar.

Un usuario que tenga un rol con un permiso de Tipo de acceso = Denegar no tendrá este permiso independientemente de si el permiso está permitido a nivel de la aplicación (de forma predeterminada) o si tiene otro rol donde el permiso está permitido.

La única forma en que se puede otorgar este permiso al usuario es con Tipo de acceso = Permitir.



Veamos escenarios en los que se realiza el control de acceso.

Primero tenemos a los Web Panel y Web Components (que solo tienen acceso desde la url - URL access = true)
En este caso se valida si el usuario tiene permiso para ejecutar el objeto. En caso de no tenerlo, no debería ver ningún dato del formulario.
Para esto, GAM verifica que el usuario tenga el permiso <prefix>_Execute, donde prefix es el Prefijo de permiso definido para el objeto.
En caso de que se detecte un error de permiso, se realizará una redirección automática al Objeto de “No Autorizado” para objetos web en el caso de ser aplicaciones web.

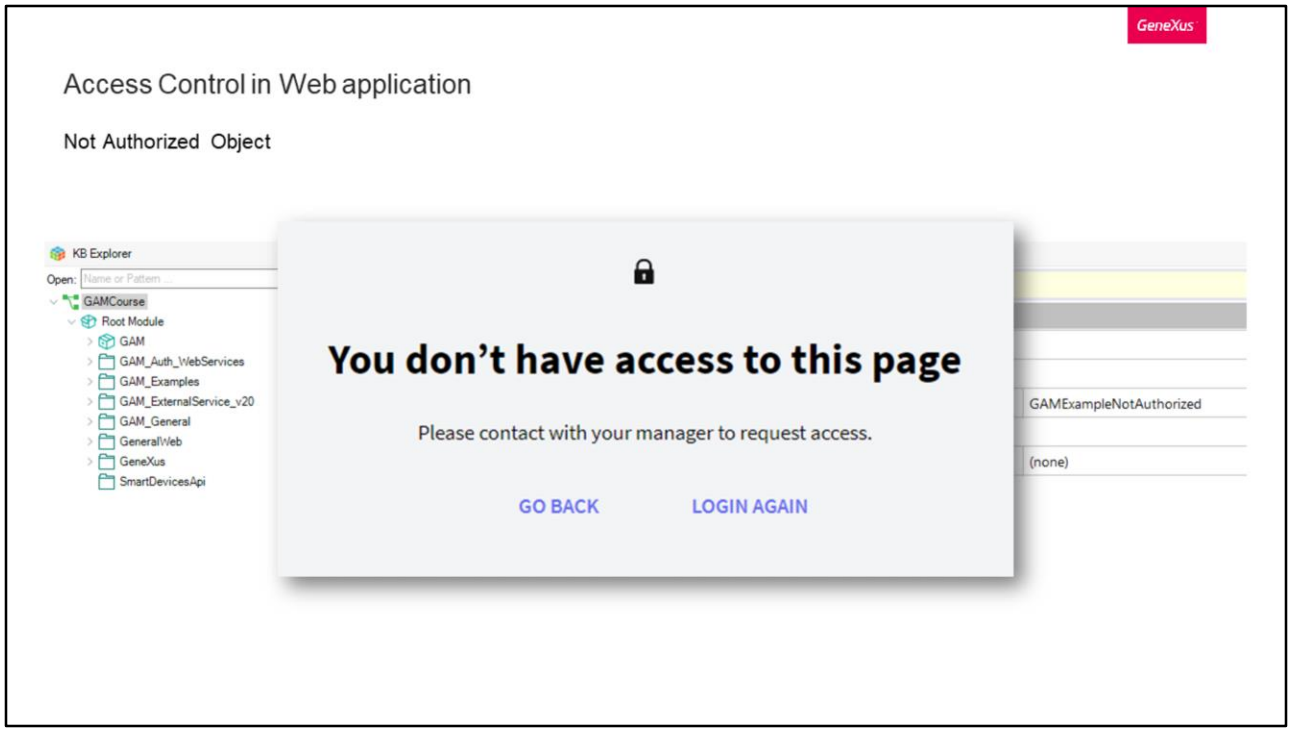
Después tenemos el acceso a un proceso principal con protocolo de llamada HTTP. Un ejemplo de esto puede ser un informe en PDF que se muestra en el navegador.
Acá se valida si el usuario tiene permiso para ejecutar este objeto. GAM verifica que el usuario tenga el permiso <prefix>_Execute, donde prefix es el Prefijo de permiso definido para el objeto.
En caso de error, se mostrará el error 401, lanzado por el servidor de aplicaciones el cual debe ser capturado por el programador.
En el caso de un informe en PDF, se considera el objeto “No Autorizado” para objetos Web.

En tercer lugar tenemos el acceso a una Transacción web con o sin modalidad.
Primero se valida si el usuario tiene permiso para ejecutar el objeto. En ese caso GAM verifica que el usuario nuevamente tenga el permiso <prefix>_Execute, donde el prefijo en este caso es el definido para la Transacción. Este permiso permite al usuario mostrar los datos de la transacción (solo en modo de visualización).
Si el usuario ejecuta una acción sobre la transacción, ya sea Confirmar o Eliminar, se requerirán otros permisos como vemos en pantalla.
De hecho también tenemos el permiso que agrupa a todos los permisos y también se puede utilizar.
En caso de error, se mostrará un mensaje de error de GeneXus si la Transacción no recibe como parámetros KEY y mode. Esto se puede ver con más detalle en la Wiki de GeneXus.

Finalmente tenemos el último punto. Acceso restringido a un grupo de páginas WEB.
Hay algunos casos donde el nivel de autorización necesario es solo para permitir o denegar a un grupo de usuarios el acceso a un conjunto de páginas web de la aplicación.

Por ejemplo, si dividimos nuestra aplicación en los módulos frontend y backend, probablemente solo algunos usuarios autorizados sean los que puedan acceder al backend.
Una forma de hacer esto es usar una Master Page para las páginas del mismo y programar lo siguiente en el evento de Start de esta.
Además se debe establecer la propiedad de la aplicación “Requerir permisos de acceso” con valor verdadero.

La otra opción es definir los permisos automáticos como hijos de ese permiso "is_authorized_toBackend" y con esto bastaría.



En la placa anterior nombrábamos mucho al objeto No autorizado. Veamos como podemos configurarlo.
A nivel de versión, podemos encontrar la propiedad Not Authorized Object tanto para Web como Mobile. Allí es donde podemos configurar que objeto de nuestra base de conocimiento es el que queremos definir para que la aplicación redirija a él cuando el usuario no está autorizado.

Por defecto tendremos que para Web se utilice el ejemplo de GAM. Se puede aprovechar este y utilizarlo, con la posibilidad de modificar su diseño.
En caso de utilizar uno propio, este debe tener configurada la propiedad de nivel de seguridad integrada en "Ninguno".

GeneXus™
by Globant

training.genexus.com
wiki.genexus.com