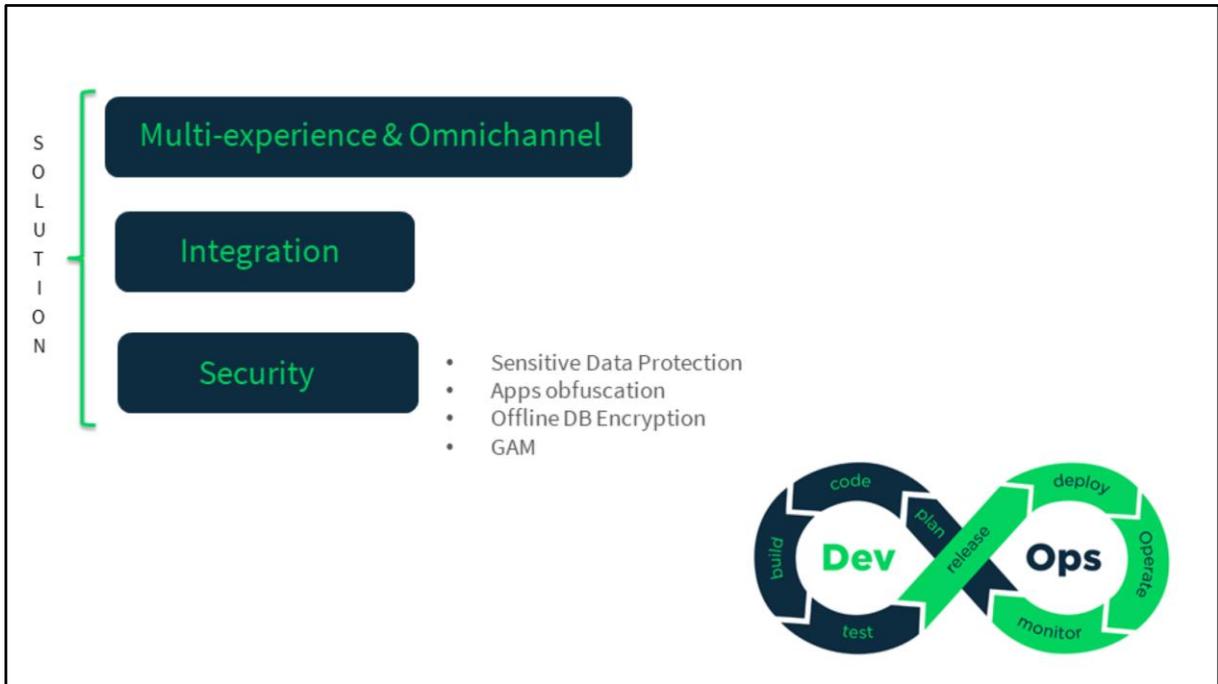


GeneXus[™]
The power of doing

Security

GeneXus™ 16



Pablo Mazzilli, nos introducía un poco en estos temas, por eso no voy a hondar en eso. Hablaba de la ofuscación, de las aplicaciones, Android, también decía que tenemos la posibilidad de encriptar la base de datos offline, entonces vamos a pasar directamente a lo que es el GAM

Data Security

Sensitive Data Protection

Apps Obfuscation

Offline DB Encryption

GeneXus Access Manager

GeneXus Access Manager

GeneXus™ 16

GAM Events Subscription

- User_Insert, User_Update, User_Delete
- User_UpdateRoles
- User_GetCustomInfo : Get user's information on the server when the [GAM Remote](#) login is executed
- User_SaveCustomInfo : This event is executed on the client when the [GAM Remote](#) login is successfully finished.
- Role_Update, Role_Delete
- Repository_Login
- Repository_Logout
- Application_CheckPermissionFail

GeneXus

Les voy a contar... se han hecho muchas mejoras en todo lo que es la versión 15 hasta ahora, hasta la versión 16, esto es un resumen.

Esto ya lo tenemos desde la versión 15, es la Suscripción Eventos, seguramente hay muchos que lo conozcan, así que no voy a profundizar mucho. Se trata de la posibilidad de poder disparar procedimientos hechos con GeneXus o no, a partir de un evento, del GAM.

Por ejemplo, el Insert de un usuario, la actualización o el delete de un usuario, en la base de datos del GAM pueden gatillar un evento programado por nosotros.

En el caso de insertar un usuario, puede servir por ejemplo, para mantener tanto en la base de datos del GAM como en la base de datos de la aplicación persistido el usuario.

Otros eventos son:

La actualización de los roles de un usuario.

Hemos introducido también últimamente en uno de los últimos Upgrades de la versión 15, la posibilidad de ejecutar eventos cuando utilizamos el tipo de autenticación GAM Remote, que se ejecutan tanto del lado del proveedor de identidades como del cliente de este proveedor.

La actualización de roles

El Login

Y el logout, también son eventos que pueden ser interesantes, digamos... que produzcan la ejecución de un evento nuestro a su vez.

Y cuando falla el chequeo de permisos, también a veces resulta interesante poder disparar un evento nuestro, por ejemplo, para advertir que a un usuario le faltan los permisos necesarios para ejecutar una acción.

Dynamic Menu

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
```

GeneXus

Bien, dado esto, sigo con otras de las funcionalidades que también ya tenemos y capaz que seguramente conozcan, que es el hecho de que podemos contar con menues dinámicos. Esto quiere decir que nosotros creamos el menú y el usuario según sus permisos y sus roles, va a tener instanciado el menú que le corresponde.

Si quedan dudas me interrumpen...

Oauth 2.0 Authentication type

- Office 365
- Mercado Libre
- Instagram
- LinkedIn
- Google
- Facebook

GeneXus

Bien...

interesante es que ahora tenemos la posibilidad de autenticarnos con un tipo de autenticación nuevo. Se llama Oauth 2.0, es genérico, quiere decir que cualquier proveedor que está basado en este protocolo pueda ser usado como medio de autenticación por nosotros, en particular:

Office 365

Mercado Libre

Instagram

LinkedIn

Google y Facebook ya contábamos con ellos, es decir, nosotros ya contábamos con Google, Facebook como tipo de autenticación que utiliza Oauth 2.0

Ahora agregamos nuevos, estos son los que estuvimos probando pero no significa que sean los únicos, puede haber otro proveedor de identidades Oauth 2.0 hacia el cual podamos nosotros conectarnos o usarlo como medio de autenticación.

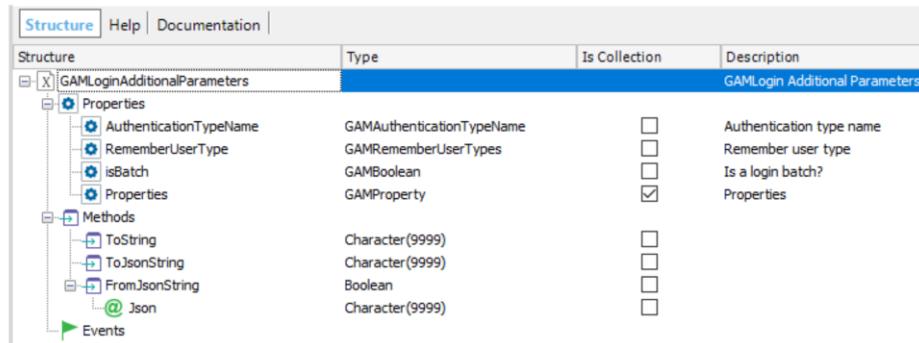
Lo importante es... como siempre, funciona como Google, Facebook y Twitter también en el sentido que ustedes tienen que crear una aplicación del lado del proveedor, se les va a dar ciertas credenciales y con esas credenciales van y configuran en el GAM ese tipo de autenticación, Oauth 2.0 con esas credenciales para poder conectarse al proveedor; Y el Login se ejecuta obviamente en el proveedor, igual que Facebook, el Login se ejecuta del lado del proveedor y él es el dueño de las credenciales del usuario.

More than one Authentication of the same type in the repository: WEB login

```
GAMRepository.LoginFacebook()
```

```
&AdditionalParameter.AuthenticationTypeName = !"Facebook1"
```

```
&LoginOK = GAMRepository.Login(&UserName, &UserPassword, &AdditionalParameter, &Errors )
```



The screenshot shows the Visual Studio Structure window for the class `GAMLoginAdditionalParameters`. The window is divided into a tree view on the left and a table on the right. The table lists the class's members, their types, whether they are collections, and their descriptions.

Structure	Type	Is Collection	Description
GAMLoginAdditionalParameters			GAMLogin Additional Parameters
Properties			
AuthenticationTypeName	GAMAuthenticationTypeName	<input type="checkbox"/>	Authentication type name
RememberUserType	GAMRememberUserTypes	<input type="checkbox"/>	Remember user type
isBatch	GAMBoolean	<input type="checkbox"/>	Is a login batch?
Properties	GAMProperty	<input checked="" type="checkbox"/>	Properties
Methods			
ToString	Character(9999)	<input type="checkbox"/>	
ToJsonString	Character(9999)	<input type="checkbox"/>	
FromJsonString	Boolean	<input type="checkbox"/>	
Json	Character(9999)	<input type="checkbox"/>	
Events			

GeneXus

Ahora tenemos la posibilidad de tener más, de un mismo tipo de autenticación en un repositorio, antes no podíamos definir, por ejemplo, dos tipos de autenticación Facebook o dos tipos de autenticación GAM Remote y ahora levantamos esa limitación.

Eso implica que seguramente vean, si ustedes ven los ejemplos del GAM, vean que el Login ha cambiado, justamente para permitir que se pueda uno autenticar, contra un mismo tipo de autenticación en el mismo repositorio.

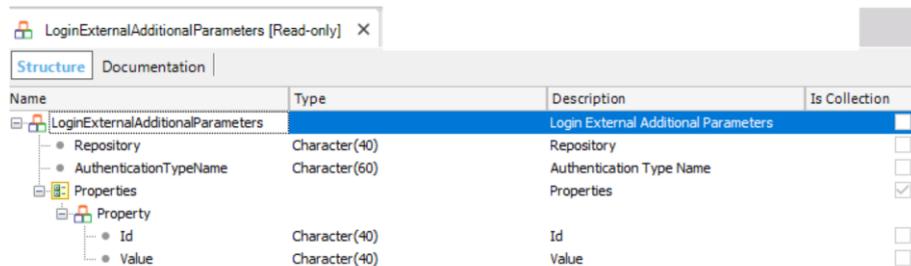
Lo que antes era el Login Facebook de esta forma, hacíamos, dado un `Repository.LoginFacebook` en aplicaciones Web, pasa a ser hoy: `GAMRepository.Login` en donde se le pasa el usuario y `AdditionalParameter`, en los Additional Parameters nosotros vamos a especificar el nombre del tipo de autenticación, el tipo de autenticación puede estar repetido, puede haber "n" Facebook en mi repositorio, "n" tipo de autenticación Facebook en mi repositorio.

En los Additional Parameters yo voy a poder especificar el nombre concreto, el nombre del tipo de autenticación al cual yo me quiero conectar, esto simplemente a modo de repaso porque seguramente ustedes hayan visto los cambios si ya tienen la versión 15, 15 upgrade 12 si no me equivoco u 11...

More than one Authentication of the same type in the repository : SD

```
GeneXus.SD.Actions.LoginExternal(GAMAuthenticationTypes.GAMRemote, &User, &Password)
```

```
&LoginExternalAdditionalParameters.AuthenticationTypeName = !"GAMRemote1"  
GeneXus.SD.Actions.LoginExternal(GAMAuthenticationTypes.GAMRemote, &User, &Password,  
&LoginExternalAdditionalParameters)
```



Name	Type	Description	Is Collection
LoginExternalAdditionalParameters		Login External Additional Parameters	<input type="checkbox"/>
Repository	Character(40)	Repository	<input type="checkbox"/>
AuthenticationTypeName	Character(60)	Authentication Type Name	<input type="checkbox"/>
Properties	Properties	Properties	<input checked="" type="checkbox"/>
Property			
Id	Character(40)	Id	<input type="checkbox"/>
Value	Character(40)	Value	<input type="checkbox"/>

GeneXus

En Smart Devices, el LoginExternal antes recibía como parámetro el tipo de autenticación. Hoy recibe como parámetro, similar a lo anterior, el nombre del tipo de autenticación, además del tipo, el nombre de autenticación para permitir esa flexibilidad que yo les decía recién.

Es decir que van a tener en la base de conocimientos, objetos nuevos, SDT nuevos, que son los que permiten pasar como parámetros el nombre de tipo de autenticación. Importante es que, en realidad la forma anterior de ejecutarse sigue siendo compatible... el Login anterior sigue siendo válido por compatibilidad, o sea, no tienen ustedes que cambiar los objetos, la programación de los objetos.

¿Hay alguna duda?

- Una pregunta, ¿hay forma de pasarle el repositorio?
- Sí, sí! A continuación yo les voy a contar.
- Está bien, gracias.
- Sí, porque justamente aprovechamos que tenemos nuevos SDT como parámetros, que incluyen elementos como parámetros al Login, para además de soportar más de un mismo tipo de autenticación por repositorio, soportar el hecho de pasar otros parámetros como el repositorio.

Establish the GAM Repository to connect to, from an SD app

```
Event 'GXLogin'  
  Composite  
    GeneXus.Common.UI.Progress.ShowWithTitle("Connecting...")  
    &LoginExternalAdditionalParameters.Repository="!"1e89a9ca-bc52-482b-a344-c4cda4a9cc8f"  
    GeneXus.SD.Actions.Login(&User,&Password,&LoginExternalAdditionalParameters)  
    GeneXus.Common.UI.Progress.Hide()  
  Return  
EndComposite  
EndEvent
```

GeneXus

...Y es esto justamente, esto es lo que responde tu pregunta...

LoginExternalAdditionalParameters es ese parámetro que yo les contaba y tenés acá la posibilidad de pasarle el "going" del repositorio.

Les explico... en las aplicaciones Smart Devices, siempre tuvimos y seguimos teniendo obviamente la posibilidad de que sean multi empresas, es decir que se puedan conectar a mas de un repositorio. Para eso lo que tenemos en el servidor es un archivo que se llama Connection.gam, con una conexión a cada uno de estos repositorios que corresponde a cada una de las empresas. Antes, lo que hacíamos para soportar este ambiente era, antes de hacer el Login, hacer un Set connection el repositorio que corresponde según la empresa. Hoy eso sigue siendo válido, por supuesto, pero además podemos establecer en el momento de hacer el Login, el repositorio al cual nos queremos conectar, que es el repositorio que corresponde a la empresa sin necesidad de pasar por el procedimiento que hace el Set connection.

Rest services in multi-company architecture

```
client_id=ad80c07c0a1046029c0655cdd9d99493&grant_type=password&scope=FullControl&username=adminfull&password=adminfull&additional_parameters={"AuthenticationTypeName":"local","Repository":"1e89a9ca-bc52-482b-a344-c4cda4a9cc8f"}
```

SAC 43563

GeneXus

Y vinculado con lo anterior, también soportamos (que esto sí es nuevo, antes no lo soportábamos) es tener: servicios Rest en un ambiente multi-empresa. Antes no podíamos llamar a servicios Rest en un ambiente donde hubiera mas de un repositorio.

No había forma de hacer un llamado, no había forma de hacer el post a un servicio Rest para que se conectara a una u otra empresa, a uno u otro repositorio.

Hoy a raíz de los cambios que yo les preguntaba anteriormente eso es posible, porque en el llamado ya pueden incluir el repositorio al cual conectarse y el repositorio que corresponde a la empresa a la cual nos queremos conectar.

SSO scenario modeled with only one GAM database for all clients

All clients share the same GAM database

All clients and Identity Provider share the same GAM database

Use Impersonate (SAC 43517)

GeneXus

Esto es importante porque muchos lo han pedido, que es el hecho de que en aplicaciones donde tenemos un ambiente de single sign-on (en ambientes de single sign-on tenemos una identity provider y clientes).

Necesariamente estos clientes tienen que tener un GAM y el identity provider tiene su propio GAM.

Antes estábamos restringidos a que cada uno de los clientes se conectara a un GAM independiente, a un GAM por sí mismo.

Ahora es posible que todos los clientes utilicen el mismo GAM, la misma base de datos del GAM ¿me explico?

Y esto es a raíz justamente de lo que yo les decía anteriormente.

Como levantamos la limitación de que se tuviera que tener un único tipo de autenticación por repositorio del mismo tipo, por ejemplo: solo podíamos tener un GAM Remote por repositorio, esa limitación obligaba necesariamente a que hubiera que tener un GAM para cada cliente. Hoy, como se puede tener varios tipos GAM Remote por repositorio, entonces podemos tener la misma base de datos del GAM para todos los clientes en nuestro ambiente de Single Sign-on.

- ¿Esto reemplazaría el escenario de Single Sign-on con GAM Remoto? Porque se podía hacer... o sea, se podía tener un GAM con una única base de datos, los clientes no tener habilitado el GAM y hacer Single Sign-on contra ese GAM.
- No, no es así. Los clientes tienen que tener un GAM, esto es por como funcionan, básicamente como funciona es: el cliente, cuando se loguea llama al identity provider, el Login se produce en el identity provider, ahí mismo porque ahí están los usuarios, etc. Pero después vuelve al cliente y en el cliente se persiste al usuario, sin la contraseña pero se persiste porque la sesión y el usuario están dentro del cliente. O sea, lo que quiero decir es, no puedes prescindir de un GAM del lado del cliente.
- Pero hay un escenario en la Wiki que lo tengo implementado, donde no tengo habilitado la seguridad integrada en las KB cliente y tengo un GAM Remoto y se implementa el Single Sign-on de esa manera, de hecho el Login se hace en el GAM Remoto, después vuelve a la aplicación cliente, obviamente hay que programar a mano algunas cosas, o sea, estaríamos reemplazando eso, ahora sería habilitar el GAM en cada cliente y sería automático ¿esa es la idea?
- A ver si te entiendo... hasta ahora era necesario habilitar el GAM en cada cliente como decís tú, lo que hicimos...
- No no, es que justamente no... hay un escenario donde se puede hacer un Single Sign-on con clientes que no usen GAM, que pueden ser para aplicaciones, por ejemplo, no GeneXus, que es como lo tengo

implementado yo, por eso digo, ahora lo podría hacer de esta forma...

- No, no! Porque es necesario, sigue siendo necesario que el cliente tenga un GAM, lo que hoy estamos introduciendo es la posibilidad de que todos los clientes usen el mismo GAM, antes, si vos tenías diez aplicaciones, cada aplicación tenía que tener su GAM, su base de datos de GAM, lo cual era un estorbo porque había que mantener diez GAM.
- Está bien, es lo que yo digo, o sea... yo tuve que implementar un GAM Remoto y programar a mano algunas cosas, ahora habilitando en los clientes GAM y diciéndole que utilicen la misma base de datos desde el GAM Remoto, podría implementarlo de esa manera, a eso voy.
- Esta bien, te entendí... ese es el segundo escenario. Esta bien, es usar una sola base de datos, el identity provider que use su GAM y que todos los clientes usen el mismo GAM que el identity provider, correcto!
- A eso me refería!

Acá siempre necesariamente usamos la funcionalidad de Impersonate, no se si saben que significa, Impersonate es tener la posibilidad de que tengamos dos tipos de autenticación pero que siempre se instancie una de ellas.

Impersonate se usa, por ejemplo, si queremos que dentro de nuestra empresa estando en la Intranet, los usuarios se autenticquen de forma local, pero en internet, por fuera de la empresa se autenticquen por otro medio, puede ser por un Web Service, por ejemplo.

Lo importante es que queremos que siempre sea un usuario, un único usuario y entonces cuando se autentican por Web Services, ese usuario "impersona" al usuario local, siempre termina siendo un usuario local, eso es Impersonate.

Y en este tipo de escenarios que estábamos hablando contigo también, es necesario usar el Impersonate. Entonces... si tenemos la base de datos del GAM con "n" tipo de autenticación GAM Remote, tenemos que seleccionar un tipo de autenticación GAM Remote, un tipo de ellos para que todos los usuarios sean impersonados con ese tipo de autenticación.

Realmente es complicado entender así al vuelo pero esta todo bien documentado y cualquier duda, yo estoy a las ordenes, pero está todo bien documentado en el Wiki

Era simplemente el objetivo para que vean las cosas nuevas que tenemos y las nuevas posibilidades que tenemos ahora. Pero el detalle en si, es complejo de explicar a fondo realmente acá

Anonymous user

Works when registration is:

FB, Twitter, Google, GAM remote & Oauth 2.0

<https://wiki.genexus.com/commwiki/servlet/wiki?19909,Auto-register%20anonymous%20user%20-%20How%20it%20works>

SAC 43508

GeneXus

Con respecto al usuario Anonymous (el usuario anónimo también ya saben lo que es) les cuento rápido. El usuario Anónimo se usa en Smart Devices y nos da la posibilidad de que el usuario pueda entrar como anónimo, justamente, pero cuando se registra... que se registra quiere decir que se loguea por primera vez, toda la información que había quedado asociada al usuario anónimo, se incorpore al nuevo usuario que se esté registrando. Antes estábamos limitados a ciertos tipos de autenticación, ahora, este también fue bastante pedido. Un usuario anónimo, cuando se registra con tipo de autenticación Facebook, Twitter, Google, GAM Remote o Oauth 2.0 va a funcionar también, es decir, va a incorporar toda la información que estaba en el usuario anónimo, va a quedar incorporada en este nuevo usuario.

Impersonate is extended for other types of authentication

Have two different authentication mechanisms but both converge on the same user.

Facebook, Google, Twitter, GAM Remote and Oauth 2.0

SAC 43466

GeneXus

Impersonate también se extiende a nuevos tipos de datos, antes no podíamos impersonar con algunos tipos de autenticación, ahora sí es posible: Facebook, Google, Twitter, GAM Remote y Oauth 2.0

GAM deploy tool for Unix environments

- Distributed with GeneXus
- Command line
- Import / Export / Connection.gam update
- Takes the DB connection of the local environment

- [https://wiki.genexus.com/commwiki/servlet/wiki?37764,GAM+deploy+tool+command+line+%28windows+and+unix-like+operating+systems%29,](https://wiki.genexus.com/commwiki/servlet/wiki?37764,GAM+deploy+tool+command+line+%28windows+and+unix-like+operating+systems%29)

GeneXus

Por último, la GAM deploy tool, ésta también es una funcionalidad que se ha solicitado bastante, que es la posibilidad de ejecutarla en ambientes que no sean Windows, hasta hoy teníamos la GAM deploy tool command line y una con interfaz gráfica que solo funcionaba en Windows, hoy tenemos una GAM deploy tool sin interfaz, es Command line que funciona en Windows, por supuesto y funciona en ambientes Unix que nos permite exportar datos, importar datos, crear el “connection.GAM”, actualizar el “connection.GAM”, no permite hoy la creación de la estructura de las tablas, eso lo haría el DBA por ejemplo, a través de scripts.

S
O
L
U
T
I
O
N

Multi-experience & Omnichannel

- Powering analytics
- BPM

Integration

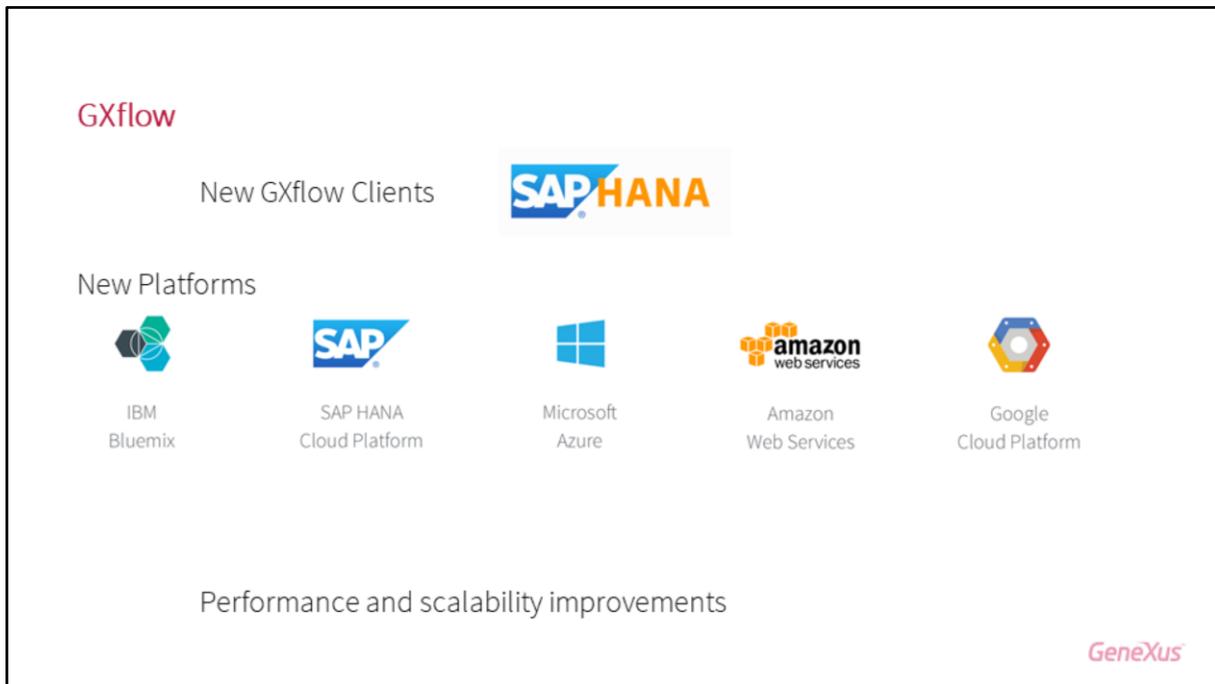
Security

- Sensitive Data Protection
- Apps obfuscation
- Offline DB Encryption
- GAM



BPM

GeneXus™ 16



Les voy a contar algunas de las cosas nuevas que tenemos en la parte de WorkFlow.

Como ustedes saben, nosotros el cliente de WorkFlow tenemos clientes que dependen de los generadores y dependen también de los DBMS, entonces en esta nueva versión agregamos un nuevo cliente, que es un cliente JAVA para SAP HANA, es decir que podemos generar ahora para esta plataforma.

Además de eso, en esta versión, el WorkFlow ahora se puede instalar como plataforma de servicios en los PAS, los que estamos soportando son: Bluemix, SAP HANA Cloud Plataform, después el de Microsoft, el de Amazon, el de Google, es decir, hasta ahora esta opción no la teníamos, no podíamos instalar ningún PAS, pero ahora a partir de esta versión ya la van a poder instalar en un PAS.

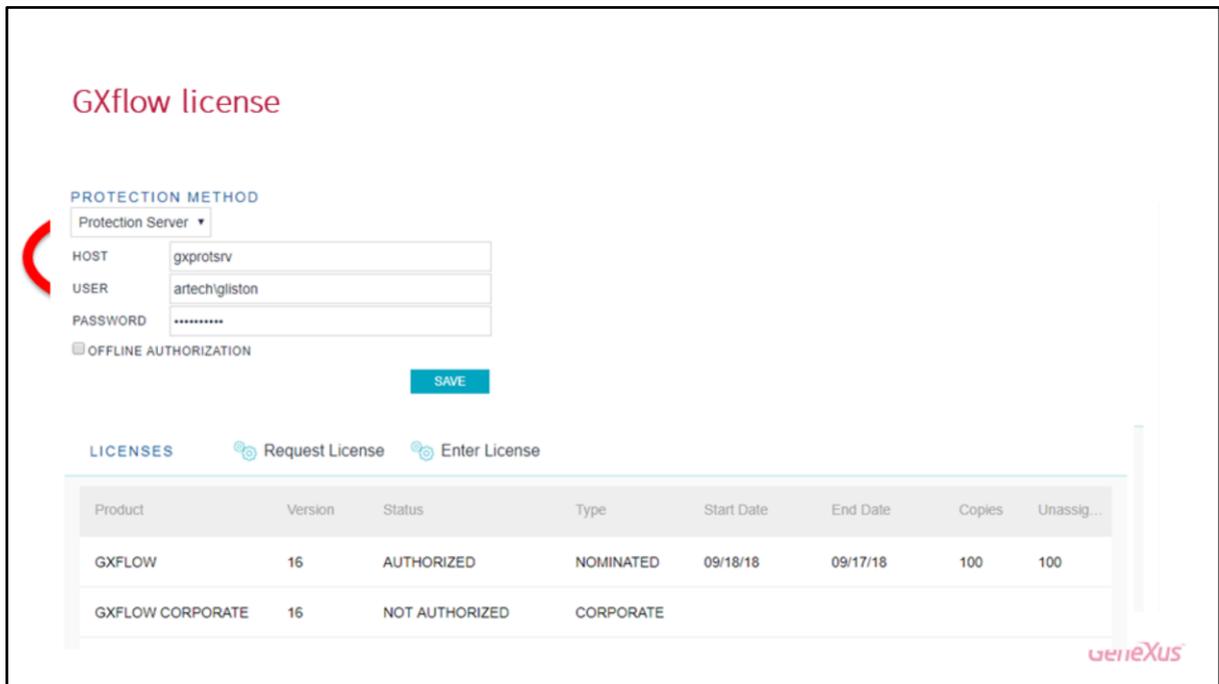
Y otra de las cosas que tiene de WorkFlow es, como ustedes saben, el cliente de WorkFlow es una aplicación generada con GeneXus, como es una aplicación generada con GeneXus tanto para Java como para NET, dependiendo lo que ustedes utilicen, al estar en la versión 16, están adquiriendo todas las ventajas y todas las cosas que ya les contaron en estos días de las ventajas que hay en Web, de escalabilidad y todo lo que ha mejorado lo van a tener dentro de esta versión.

En especial ¿Por qué se los quiero recalcar?

Porque algunas de las cosas que hemos cambiado y que hemos mejorado es: los paginados, es algo de que (creo que ya lo vieron con Javier) ahora tenemos el paginado en el Server, entonces el WorkFlow lo empezó a implementar en todas las partes que tiene paginado, las páginas donde uno va viendo los Work Items, es decir, todas las tareas que tienen los usuarios empiezan a tener paginados con el server.

Y otra de las cosas que nos ha ayudado mucho a mejorar la performance es que se agregó toda la parte del "Caching" ya había muchas cosas de caching en GeneXus, ahora en la 16 se mejoraron y WorkFlow las empezó a utilizar.

Tenemos "cache" en la parte de las tablas, en la parte de las sentencias estamos utilizando, tenemos caches para ciertos parámetros y para ciertas sesiones, entonces todo eso ha ayudado a que la herramienta ahora funcione mucho mejor cuando tenemos mucha mas carga de datos, lo hemos mejorado bastante.



Se cambió la protección, como ven ahora, esta es una aplicación web.

Las licencias de WorkFlow siempre se manejaban con Protection Server, License manager como cualquier otro de los productos de GeneXus y nosotros ahora, dentro del cliente de WorkFlow, lo que se agregó es un License manager, es un License manager Web, quiere decir que no tienen porque ir al License manager de antes, le pueden manejar todas las licencias desde aquí.

Otra de las cosas que se hizo es, las licencias de WorkFlow, teníamos licencias nominadas y licencias corporativas, nominadas es decir, comprábamos para determinada cantidad de usuarios, teníamos que decir quienes la usaban. Y la corporativa es que era una licencia y entraban indefinidos usuarios, pero teníamos un único producto en el licenciador, entonces ahora lo que hicimos es, lo separamos, tenemos uno para las nominadas, otro para las corporativas, entonces lo pueden pedir de forma independiente.

Y otra de las cosas que cambiamos es, si vienen aquí, tenemos que la protección tiene dos opciones: Native o Protection Server.

Protection Server sigue siendo las mismas licencias de antes, que las pueden manejar desde este License manager o de los License managers anteriores

Y La Native, lo que estamos haciendo es cambiando en vez de guardar la protección de la forma en que la guardábamos antes, ahora la estamos guardando en la base de datos de WorkFlow, es decir, una licencia que va a nivel de base de datos.

¿Qué es lo que esto nos permitió? Nos permitió poder instalar las aplicaciones de WorkFlow en los PAS, porque ustedes no tenían la opción de poder instalar toda la parte de las licencias, las de protección en el servidor, entonces ahora cuando hacen el deploy y suben la aplicación, a partir de la interfaz web piden la licencia, la licencia queda guardada a nivel de base de datos y eso le funciona, eso es lo que nos permitió hacer lo que antes les comenté.

Entonces, cuando seleccionamos que queremos la protección “nativa” se nos habilitan aquí dos opciones, el servidor del cual lo están instalando, que muchas veces no tiene conexión a internet que eso nos pasa en los servidores donde esta la aplicación, pueden marcar que es offline, entonces va a generar el archívito, el GXA para poder pedir la licencia, ahí la van a ingresar, lo mismo que hacíamos antes.

Y lo otro es que tenemos la opción de generar LOG, lo mismo que teníamos en el License manager, una opción para generar LOG para cuando a veces, todos conocemos que tenemos algún problema con las licencias, algo no está funcionando, podemos generar el LOG desde esta opción.

Si seleccionamos la opción de protection server, lo que vamos a tener que hacer es lo mismo que hacíamos, tenemos que tener algún servidor donde este el servidor de protección y las licencias instaladas, eso sigue siendo lo mismo que antes, lo único que tenemos que colocar es que hay que decir cual es el nombre del servidor, el usuario y la contraseña con la cual nos vamos a estar conectando.

Tienen que colocar acá, es un detalle, es: dominio/usuario, no es solo usuario, es dominio y usuario, tiene que ser un usuario que tengamos permiso de acceso a las licencias que están en ese servidor. Y como ven acá, tenemos en este servidor 100 licencias nominadas que recién las acabamos de instalar, estaban sin asignar, así que están todas las licencias prontas para empezar a usar nosotros dentro de nuestro sistema.

Así que este fue un cambio un poco revolucionario, quienes utilizan WorkFlow seguro le van a ver bastantes ventajas de poder manejar las licencias desde aquí y a su vez tener las licencias a nivel de base de datos que es algo que nosotros creemos que va a ser donde mucha gente lo va a comenzar a utilizar.

GXflow internally uses the Log external object to write low level log information

```
....  
11:18:04,768 [1] DEBUG GeneXus.Data.GxSqlCursorDataReader - Return LoadFirstBlock, Parameters: CursorHandle 180150009  
11:18:04,768 [1] DEBUG GXFLOW.STATE_MACHINE - Evaluate outgoing sequence flow  
11:18:04,768 [1] DEBUG GXFLOW.STATE_MACHINE - Task: 27  
11:18:04,768 [1] DEBUG GXFLOW.STATE_MACHINE - Flow Type: F  
11:18:04,768 [1] DEBUG GXFLOW.STATE_MACHINE - Create outgoing task : 27  
11:18:04,772 [1] DEBUG GXFLOW.STATE_MACHINE - Create Workitem  
11:18:04,772 [1] DEBUG GeneXus.Data.ADO.GxCommand - GxCommand.FetchSize: 1, handle '1'  
....
```

Y de las otras cosas que creo que mas recalcales tiene también, es:

En estos días ya vieron, creo que fue con Armin, que vieron el tema de que ahora las aplicaciones GeneXus ustedes cuando las programan pueden generar LOGs de su propia aplicación.

El WorkFlow como es una aplicación GeneXus como hablábamos antes, también hizo lo mismo, empezó a generar logs de todas las actividades.

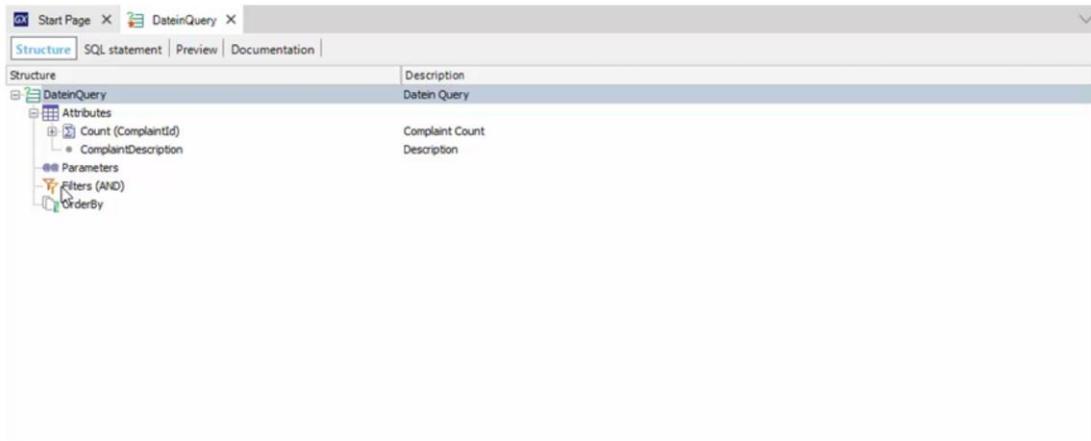
Entonces ahora, cuando ustedes utilizan las “API” de WorkFlow y todo eso que para ustedes era una cajita negra, que no sabíamos cuales eran las acciones y no podíamos “debuggear” a veces no sabíamos que es lo que pasaba internamente, si habilitamos el log de GeneXus con las opciones más altas, que es la de “debug” y la de “All”, ahí empieza a grabar el WorkFlow información en el log de la aplicación, entonces, eso nos va a permitir a todos darnos cuenta cuales son las cosas que están sucediendo, a veces somos nosotros que estamos colocando mal el tipo de datos, no le estamos pasando el valor correcto de la instancia del proceso, o el código del usuario, entonces nos va a ayudar a mejorar nuestra propia programación, es decir, dejó de ser una cajita cerrada y vamos a poder tener esa interacción para mejorar. Así que a grandes rasgos, las grandes funcionalidades y algo que quería comentarles es, el cliente de SD no se los mostré, en realidad lo seguimos teniendo, sigue teniendo la misma interfaz y los invito a empezar a utilizarlo.

En la web hay una KB que se pueden bajar desde GeneXus Server o hay un XPZ para poder empezar a hacer los WorkFlows integrados, que tengan parte de aplicación web y parte de aplicación SD que tiene sus ventajas, es decir, no ha cambiado la interfaz, sigue siendo la misma pero se los recomiendo empezar a usar en sus aplicaciones.

Analytics: Reporting and Dashboards

GeneXus™ 16

&Today en el Query



La parte de Reporting y Dashboards...

Entonces, les voy a contar un poco:

Hasta ahora veníamos teniendo el objeto Query como siempre, lo que hemos hecho en estas últimas versiones es agregar mucha más funcionalidad al objeto Query, una de ellas es poder tener dentro del objeto Query la utilización de la variable &today.

Parece algo sencillo, pero hasta ahora no lo teníamos, entonces acá podemos tener una consulta en la cual ustedes venían trabajando con la KB de las quejas, de las personas que hacían en la ciudad, entonces lo que hacemos es, queremos hacer es ver la cantidad de quejas por descripción pero que hay hasta la fecha de hoy.

Entonces, lo único que hay que hacer es en la parte de los filtros, ponemos la fecha y le decimos “menor o igual que &today” y ahí la consulta lo que hace es, busca todas las quejas que son menores que la fecha de hoy y automáticamente se está ejecutando esa consulta.

Ahí está la fecha, dice que la fecha tiene que ser \leq &Today (menor o igual a la variable &Today) y automáticamente cuando pasemos al Preview, si se fijan, el número va a cambiar, porque estaba mostrándome todas las quejas que había hasta esa fecha y ahora son hasta al fecha del día de hoy, 47 y ahí pasan a 26.

Si vieron esa interfaz del Objeto Query que se mostró en una tabla, eso también cambió en esta versión, se modificó toda la estética de la salida de las gráficas, de las salidas de las pivottables, se cambió esa estética, también el menú que tiene para poder mostrar para exportar las consultas, toda esa estética se ha cambiado, ha mejorado bastante.

Y algo que es bastante útil para todos nosotros, esa misma estética hoy en día está enganchada con el objeto Tema, ahora se le pueden asociar las clases que están en el objeto Tema o ustedes se pueden crear sus propias clases para cambiarle la estética al Objeto Query, entonces ahora queda todo integrado dentro de la aplicación, podemos ponerle una estética mejor a todas las consultas aplicando lo que ya tenemos dentro de nuestra KB.



Otra de las cláusulas bastante interesantes que se agregaron ahora en este objeto, es poder decir “utilizar la clausula For each” dentro del Objeto Query.

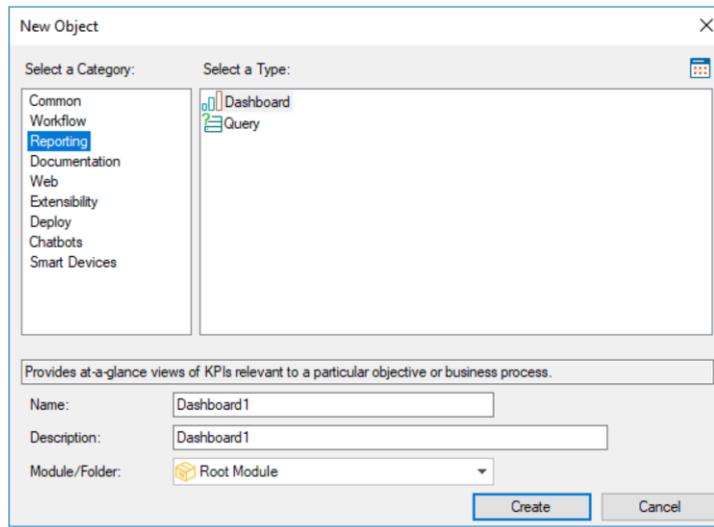
Acá lo que vamos a tener es un indicador, que nuestro indicador por ejemplo es la cantidad de quejas y lo que vamos a hacer es analizarlo por una dimensión diferente.

Esta consulta (en realidad la original) lo que nosotros teníamos era la cantidad de quejas por año, entonces agarraba y me decía la cantidad de quejas que teníamos en el 2017, en el 2018, pero yo dije “no, no quiero ver solo eso”, quiero ver la cantidad de quejas por año pero desglosado por los tipos de quejas, entonces simplemente lo que hago es “For each/ cuál es el tipo de queja”.

Eso lo que hace es esta consulta, lo que me va a mostrar es esto que se ve acá, para el año 2017 y 2018 dependiendo de los tres tipos de quejas que tenemos, que son la de los Parques, señales de tránsito y los semáforos, entonces está mostrando que acá tenemos 4, acá tenemos 10 y acá tenemos 3 y me va haciendo ese desglose, es decir, lo que estamos colocando es un único indicador, pero en realidad lo estamos dividiendo por diferentes dimensiones.

Así que esta es una gran potencia que nos da a nosotros ahora en el Objeto Query para hacer consultas más complejas.

Dashboard Object



The screenshot shows the 'New Object' dialog box in GeneXus. It has two main sections: 'Select a Category:' and 'Select a Type:'. Under 'Select a Category:', the following options are listed: Common, Workflow, Reporting (highlighted in blue), Documentation, Web, Extensibility, Deploy, Chatbots, and Smart Devices. Under 'Select a Type:', the following options are listed: Dashboard (with a dashboard icon) and Query (with a query icon). Below these sections, there is a text box containing the description: 'Provides at-a-glance views of KPIs relevant to a particular objective or business process.' At the bottom, there are three input fields: 'Name:' with the value 'Dashboard 1', 'Description:' with the value 'Dashboard 1', and 'Module/Folder:' with a dropdown menu showing 'Root Module'. There are 'Create' and 'Cancel' buttons at the bottom right.

Ahora vamos a pasar al objeto principal que se creó en esta versión, es el Objeto Dashboard (es un objeto nuevo que lo van a encontrar en a parte de Reporting), además del Query van a tener el Dashboard.

Los Dashboard seguramente ustedes ya los conocen, es una pantalla en la cual uno tiene diferentes consultas con diferentes indicadores, lo que va haciendo es modificando filtros y lo que nos va mostrando es una imagen de cómo está nuestra empresa y si sirve para poder tomar decisiones dentro de la misma.

Hasta ahora todo eso que yo les cuento, “ya lo hacíamos” me van a decir ustedes, teníamos antes el Objeto WebPanel, los Objetos query y lo que hacíamos era colocar los diferentes objetos Query y mediante programación podíamos tener esa lógica.

Ahora lo vamos a hacer, pero lo vamos a hacer de una manera mucho más fácil, en vez de tener que estar programando lo vamos a hacer de manera declarativa.

Dashboard Object

- Not programming required
- Queries, Filters, Textblocks, Images
- Live editing
- Not executable object

¿Cuáles son las principales ventajas que tiene el Objeto Query?

No van a requerir programación, lo único que vamos a hacer es arrastrar Querys dentro del objeto Dashboard, definir algunas propiedades dentro de cada una de esas Query, les vamos a decir algunas configuraciones y a partir de ahí vamos a tener nuestro propio Dashboard.

¿Qué es lo que vamos a poder poner dentro del Dashboard?

Vamos a poder poner Querys, vamos a poder poner filtros, vamos a poder poner Textblocks (es decir, vamos a poder poner textos para poder ordenar y que queden más bonitas las gráficas y que tenga algún significado, gráficas o Querys).

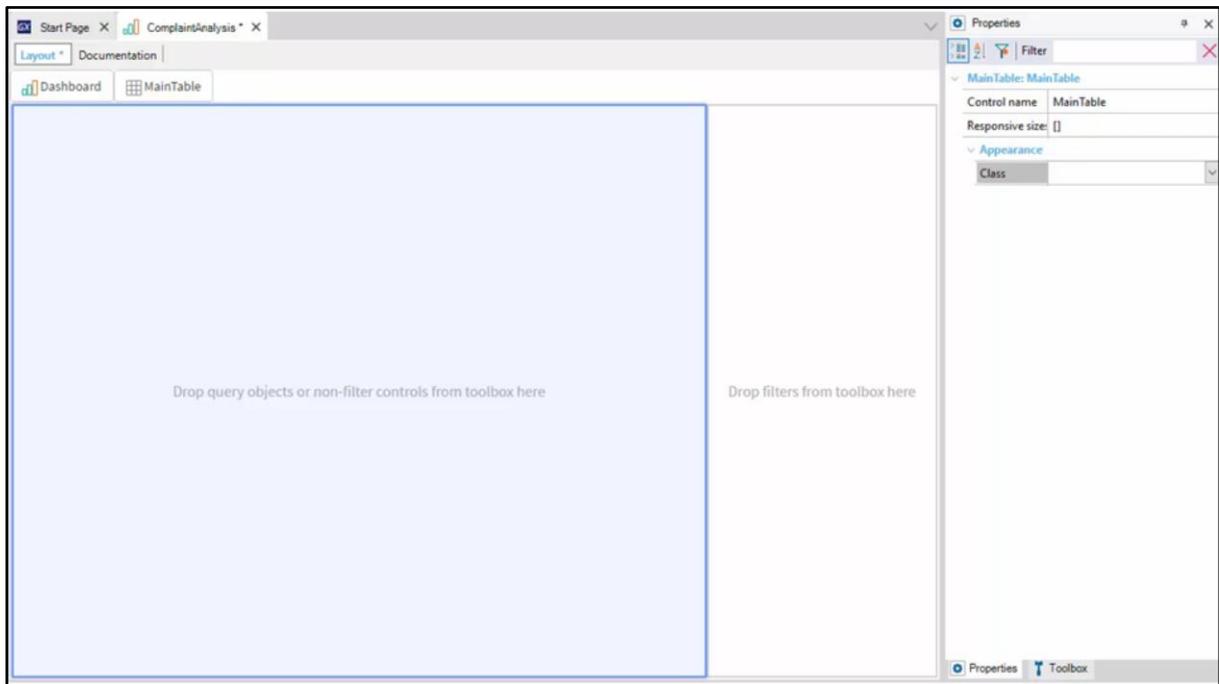
Y también poner imágenes, si ustedes quieren tener logos de la empresa o alguna otra cosa que le quieran colocar, también se va a poder tener.

¿Qué es lo que va a tener? Es Live editing

Live editing, a medida que ustedes van definiendo un Dashboard y le van agregando una Query, si ustedes ya tienen la base de datos creada y si tienen datos ahí, lo van a ver de forma automática el resultado de esa consulta, entonces van a poner los filtros, al momento de cambiar los filtros las consultas se va a refrescar y van a estar viendo lo mismo que se va a ver después en ejecución, en ese momento ya están probando todo, que va a ser lo mismo que se va a ver en ejecución, la diferencia va a ser que ahí están con los datos de prueba y cuando lo ponen en producción va a ir contra la base de datos de producción, pero la estética la van a poder definir, van a poder cambiar los colores, los tipos de letra, todo y lo van a ver en el momento.

Lo que el Objeto Dashboard no es... no es un objeto ejecutable, este objeto no se ejecuta por si solo, para poder usar este objeto, lo que vamos a tener es un Dashboard Viewer, como teníamos un Query Viewer, que asociado acá es lo mismo.

Vamos a tener que crear un Web Panel, colocar el Dashboard Viewer, el User Control y le vamos a asociar el Dashboard y a partir de ahí lo vamos a poder utilizar.



... Les voy mostrando un poco cómo funciona:

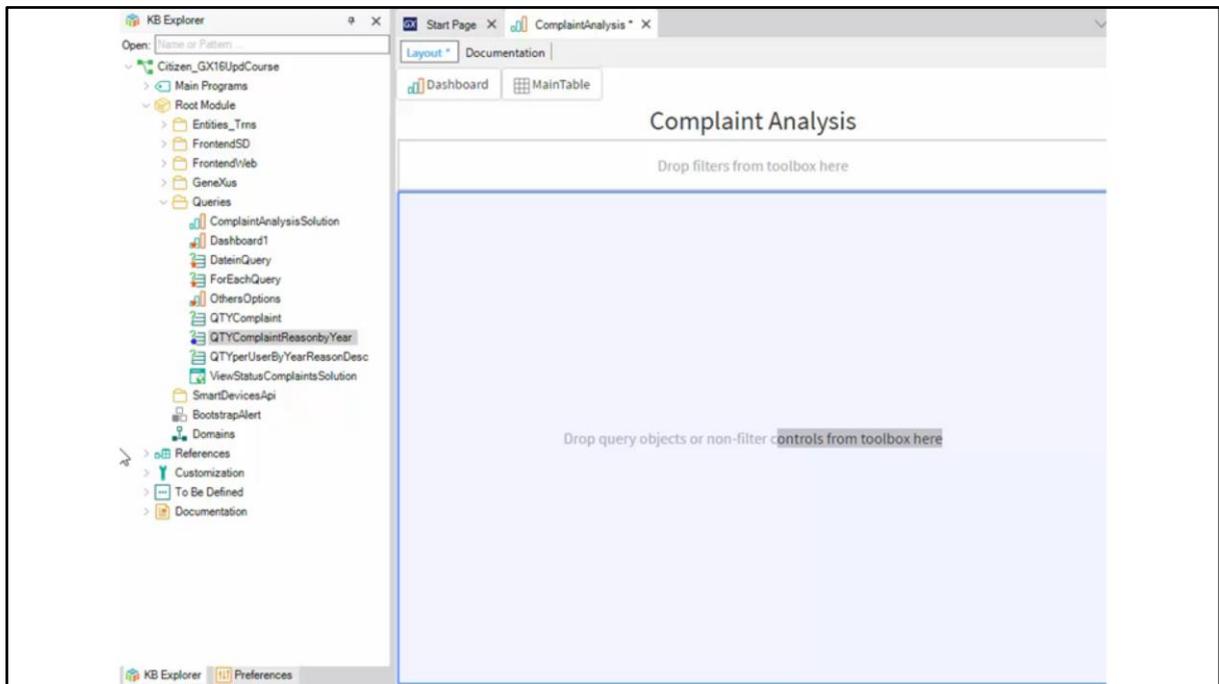
Una vez que ustedes se crean el objeto Dashboard, les va a mostrar dos áreas, un área que es para las Querys y un área que es para la parte de los filtros, en este caso tiene dos áreas nada más, pero eso se puede personalizar, ustedes pueden decirle cuantas áreas quieren tener, si quieren tener dos filas, tres, eso hay también una forma de personalizarlo.

Le pueden indicar qué es lo que quieren, si quieren que los filtros estén en la parte de arriba, si quieren que los filtros estén a la derecha, si quieren que los filtros estén abajo.

Le podemos poner un texto como yo le coloqué recién, un texto o un título, tienen toda esa funcionalidad para ir personalizando.

Para acá es la parte inicial nada más, ahí ven la forma de ponerlo que quede arriba y ahí es la forma de colocarle que tenga un título.

Tenemos una opción acá también que a este Dashboard le podemos decir cada cuanto queremos que se refresque, entonces le podemos poner un periodo y automáticamente se van a ejecutar todas esas consultas con esos datos y nos va a estar mostrando la información.

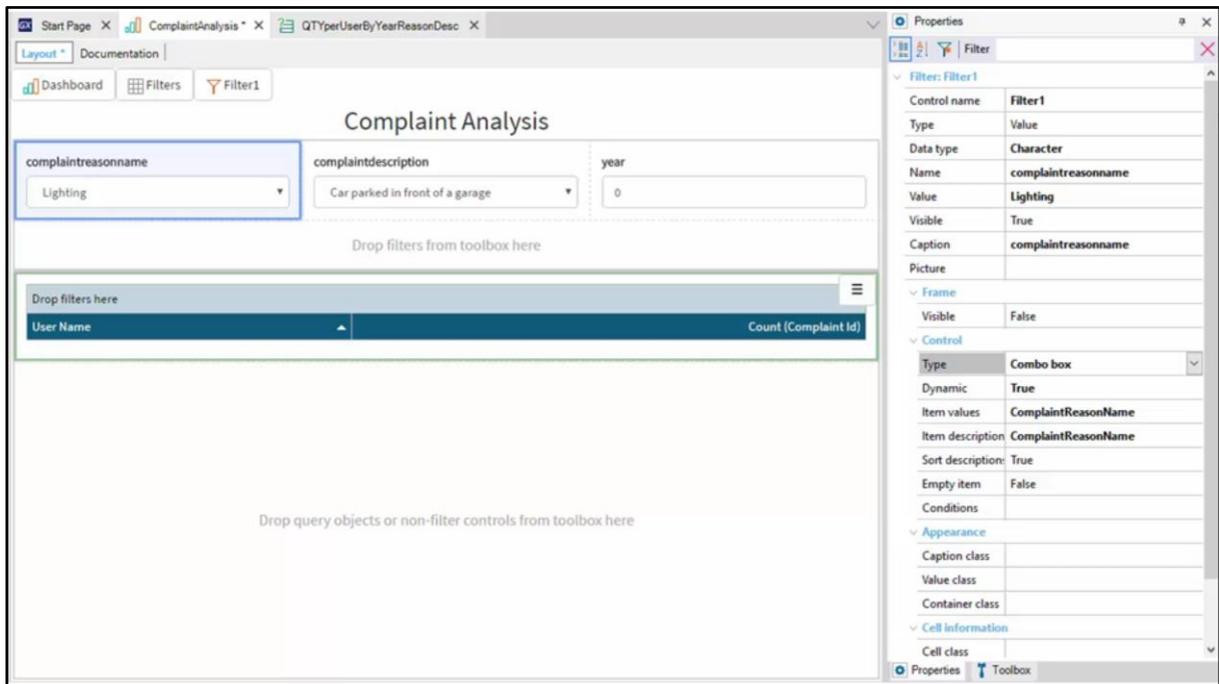


Una vez que definimos ya la primera parte, lo que vamos a tener que hacer es empezar a colocar las consultas.

Acá tengo una consulta que recibe tres parámetros, uno es el año, la descripción de la queja y la razón de la queja y simplemente lo que voy a hacer es arrastrarla al área donde van las Querys, una vez que arrastro al área que van las Querys, automáticamente GeneXus me define los parámetros porque él sabía que esa consulta recibía tres parámetros, además de eso, él automáticamente ya se dio cuenta que el parámetro ese que está ahí va sobre el atributo que era el "Reason Description", entonces ya fue y me creó que es un combo dinámico y me tomó los valores del combo dinámico, lo mismo me hizo con el otro que es los tipos de quejas que habían, si eran quejas para los parques, para los semáforos, para las señales de tránsito, para el único que no me hizo eso fue para el año, porque en realidad en mi transacción no tengo guardado el año, lo que tengo guardado es una fecha, entonces él no sabe cuáles son los valores, entonces yo voy a hacer algo diferente para poder cargar eso.

Como ven ya hay una cantidad de lógicas que ustedes no la tienen que pensar, GeneXus automáticamente ya se las identificó.

Dependiendo del tipo de datos, él les va a poner si es un Dynamic Combo, si es un Campo Edit, ese tiene cierta lógica con la cual va a haber Dynamic Como, Campos Edits, va a haber radio buttons, sliders y algunas otras opciones que ahora no me acuerdo pero tienen varias maneras diferentes de colocar los filtros, esas son algunas de las opciones.



¿Qué es lo que haríamos?

En este caso empezamos a personalizar un poco, dejarlo más “bonito” esto que se está viendo acá.

Lo que vamos a hacer es:

Nos paramos acá, ven que esté quedó azul porque es el que estoy posicionado y este esta pintado de verde (capaz que no se llega a ver) eso lo que quiere decir es que me paro en un filtro y me marca de verde todas las consultas que ese filtro aplica, en este caso tengo una sola en pantalla, cuando ustedes hagan el práctico van a tener varias y van a ver eso, que van a seleccionar un filtro y les va a mostrar cuáles son todas las consultas, entonces también cuando ustedes hacen un cambio en ese filtro, les deja la referencia para decir “bueno, este cambio que haga acá va a modificar, va a aplicar a esta, a esta y a esta” porque a veces pueden tener diferentes filtros en pantallas que no aplican a todas las consultas que tienen ustedes en el Dashboard.

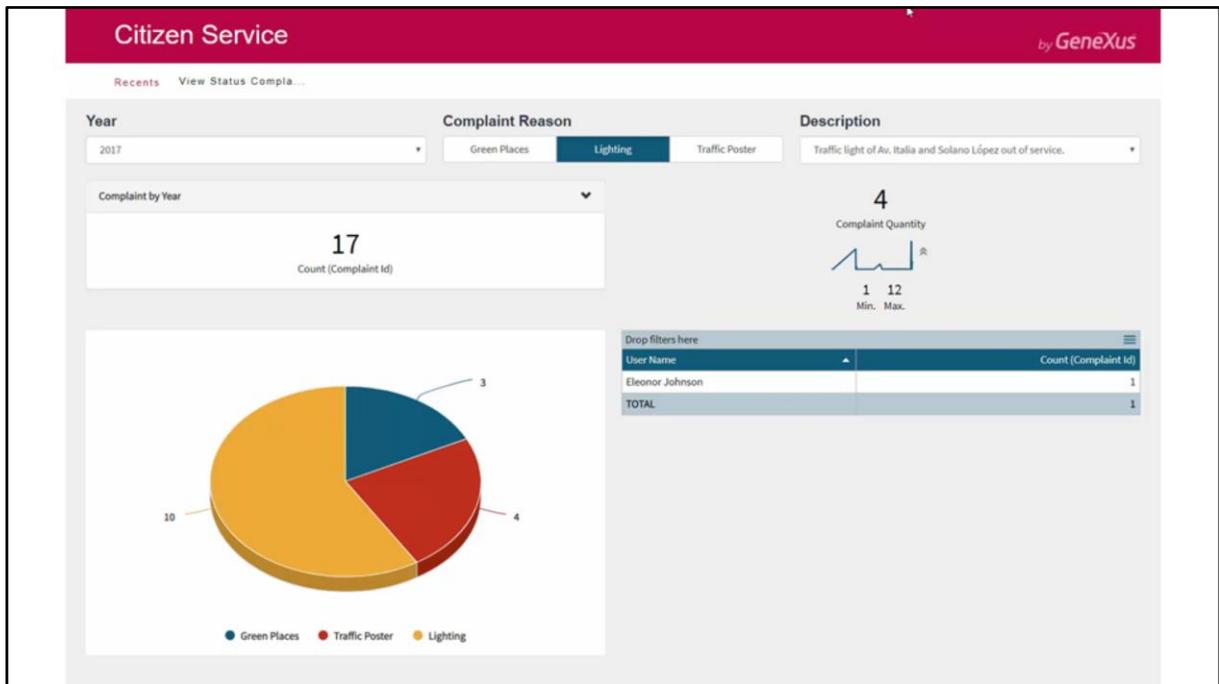
Lo único que hice acá es cambiarle el nombre porque puso por defecto la descripción que tenía ese atributo, ahora lo único que estoy haciendo es cambiando.

Ven que acá dice que es un Combo Dinámico y me toma acá directamente cuales son los valores de los cuales los va a cargar.

Lo mismo hace para éste, me dice que es la descripción, toma los valores para los cuales los va a cargar.

Y acá para el campo de Año, me lo dejo como un campo Edit, lo que voy a hacer yo es lo voy a cambiar, en vez de decirle que es un Edit y lo que voy a hacer es un Combo box y le voy a poner valores fijos, en éste caso yo decidí que van a ser solamente 2 años y lo puse manualmente, se podría hacer de otra manera, podríamos tener una tabla que sea de los años y también cargarla, pero para mostrar un ejemplo diferente.

En este caso acá (que ahora les muestro en la pantalla) si nosotros seleccionamos la razón, las descripciones tienen relación con las razones, yo acá no puse pero se puede definir una condición para que cuando seleccionemos una razón de una queja, me muestre todas las demás, solamente las que son válidas, eso en el practico lo van a ver y ahí se les explica cómo hacerlo, pero como ven, tiene mucha potencia y estamos pudiendo hacer todas estas definiciones en el momento.



Una vez definido un Web Panel en el cual le asociamos el Dashboard Viewer y le colocamos éste Dashboard, lo vamos a ver así en ejecución. Acá tenemos los años, yo cambié los ordenes, acá podemos ver el año, ésta es la razón y ésta es la descripción.

Acá lo cambié en vez de dejarlo como un combo dinámico, lo que puse es como un Radio Buttons, como en este caso tenía tres opciones y eran fijas, entonces quedaba mucho más lindo que quedara así como un Radio button, a medida que me voy cambiando de acá, me cambia estas y me va mostrando las gráficas.

Si ven, tenemos cuatro formatos diferentes acá, en realidad son todas Querys, solo que les puse que la salida era distinta.

Ésta es una salida de tipo card, ésta también es una salida de tipo card pero tiene otras características definidas, esto me está diciendo cuales son todas las quejas que hubo en este año y esto me está diciendo cuales son las quejas que hubo hasta este día nada más y cual fue el máximo y el mínimo de quejas. Me va a ir mostrando una gráfica que va a decir “tuve 30, ahora tengo 4 al día de hoy, pero el máximo que tuve en un día fue de 12”, hubo un día que se ve que se rompió un semáforo y todo el mundo empezó a reclamar y ahí me levantó la gráfica, me va mostrando la gráfica como sube y como baja.

Acá en realidad me muestra lo mismo que esto pero me lo muestra dividiéndolo por los tipos de quejas, es la misma consulta que está dos veces en pantalla con dos salidas distintas.

Y ésta simplemente lo que hace es hacerme el desglose, me dice “estas personas cuantas veces se han quejado durante el año”, cuantas veces se han reportado cosas y ahí lo que podemos ver es que hay ciertas personas que se comprometen también con la sociedad y están reportando cosas que están mal para ayudar en el bien de todo, pero eso es lo que nos permite ver esta consulta.

Y bueno, acá es lo principal del Dashboard, tiene muchas más cosas, muchas más propiedades como les decía, en realidad nos olvidamos de programar, lo único que hice fue tirar consultas en la pantalla y a partir de ahí vamos definiendo los filtros y algunas cosas.

- Hola buenos días, una consulta ¿Hay alguna opción de, por ejemplo, este Dashboard tener un botón

- que diga "PDF"? Porque generalmente me pasa que tengo consultas que los usuarios las cambian, pero ¿las puedo imprimir? Tengo que mostrárselas a mi gerente, ¿existe ese auxilio?
- Te va a contestar la persona indicada que es el desarrollador, Gustavo Martínez, se los presento.
 - Las Tablas y los Pivottables en la esquina superior derecha tienen un menú y ahí eso tiene "exportar a PDF", ¿eso es lo que buscabas...?
 - No, él quiere todo entero...
 - Ah... todo entero
 - Como una foto
 - Print Screen lo tenemos que implementar...
 - Como idea esta buena sí, la vamos a tener en cuenta pero no tiene por ahora.
 - Tengo un WorkFlow que aplica toda la empresa que es de reclamo, entonces los usuarios a nivel nacional hacen un reclamo, después pasa a la parte de administración, una parte a calidad... es algo complejo, pero la gente de calidad hace su análisis y tiene que presentar semanalmente su reporte a su jefe y me encuentro con que, claro, le tengo las gráficas, todo, pero necesito que se saque a papel...
 - Sí, tomamos la sugerencia..

 - ¿Si requiriéramos alguna programación...
 - ¿Programación de que tipo?
 - Tal vez por los filtros o un Query muy complicado...
 - Bueno pero esto es lo mismo, es un Query que en realidad puede tener un Objeto Query atrás, puedes tener un Data Provider, en eso no hay restricción, en eso seguimos siendo lo mismo de antes, cargas el Query Viewer, porque en realidad acá sería el Query Viewer, lo cargas con una Query o lo cargas con un Data provider.

GeneXus[™]
The power of doing