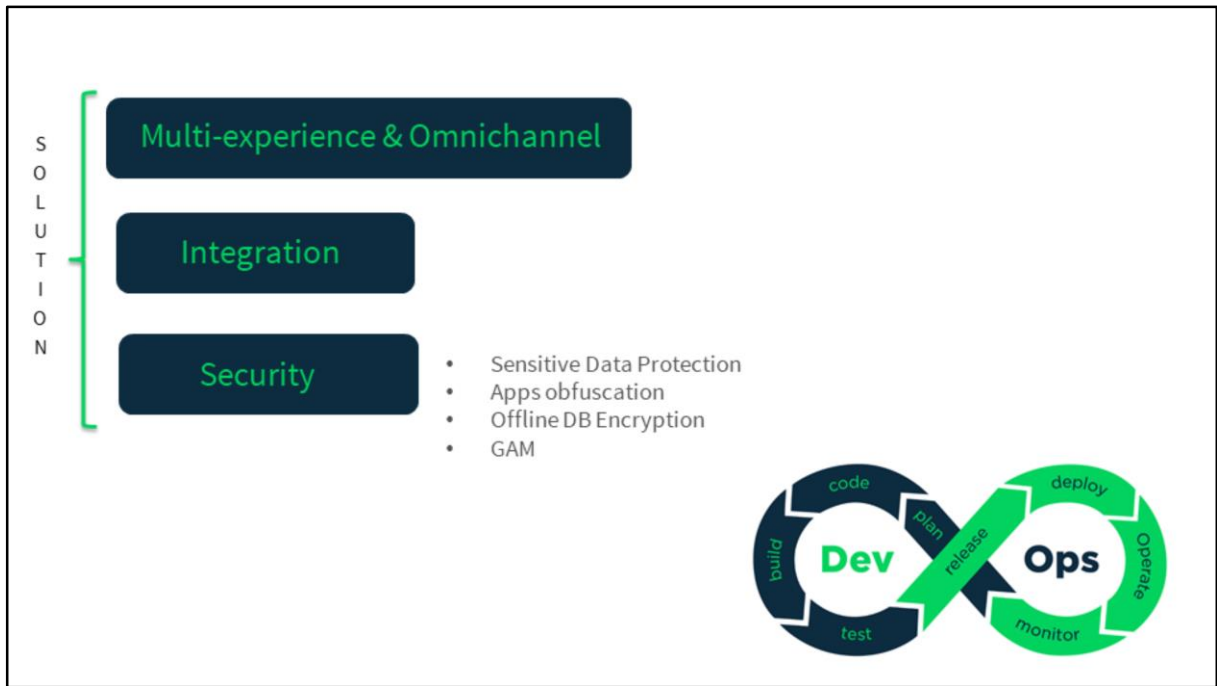


GeneXus™
The power of doing

Security

GeneXus™ 16



Ahora entraremos en los aspectos de seguridad de los datos de nuestras apps.

Data Security

Sensitive Data Protection

Apps Obfuscation

Offline DB Encryption

GeneXus Access Manager

Sobre los primeros temas ya hablamos antes, por lo que ahora nos dedicaremos a hablar específicamente sobre GAM.

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

User_Insert: Insert of a GAM User

User_Update: Update of a GAM User

User_Delete: Delete of a GAM User

User_UpdateRoles: Change of roles of a list of users

User_GetCustomInfo : Get user's information on the server when the [GAM Remote](#) login is executed. It's executed when the user logs in to the server, and also in the SSO login (when an app connects from another tab of the browser, and the session already exists so the user doesn't need to login again)

User_SaveCustomInfo : This event is executed on the client when the [GAM Remote](#) login is successfully finished.

Role_Insert: Insert of a role

Role_Update: Update of a role

Role_Delete: Delete of a role

Repository_Login: GAM User login (any [GAM Authentication Types](#) is supported)

Repository_Logout: GAM log out

Application_CheckPermissionFail : Failure of a permission verification.

Dynamic Menus

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

Using [GeneXus Access Manager \(GAM\)](#), your WEB application menu can be defined (dynamically) at runtime, based on the [GAM Permissions](#) and [GAM Roles](#) of the user logged-in.

The end user will view only the Menu options linked to resources associated with the permissions that the user may access.

The purpose is to filter the menu options according to the end user's permissions. Here, GAM returns the menu structure depending on the permissions of the user so that this structure may be loaded at runtime with any User Control.

The menu may be displayed using any User Control.

See <http://wiki.genexus.com/commwiki/servlet/wiki?29743,Howto%3A+Display+a+GAM+Menu+using+Jscookmenu+UC>,

This is an example of getting the menu options for the current user.

Oauth 2.0 Authentication type

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

GeneXus

<https://wiki.genexus.com/commwiki/servlet/wiki?39484,GAM%20auth%202.0%20Authentication%20Type>

The definition of this [GAM Authentication Type](#) is the same as any other type of authentication already existing in GAM, only that this type of authentication requires detailed configuration of the protocol used by the Provider.

At runtime, the operation is similar to the one we already know about Facebook / Google authentication, for example. When we select this type of authentication, the login is redirected to the Identity Provider configured.














The Login is displayed by the Provider; there, the user enters his/her credentials and then is redirected back to the application.

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 )
```

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			

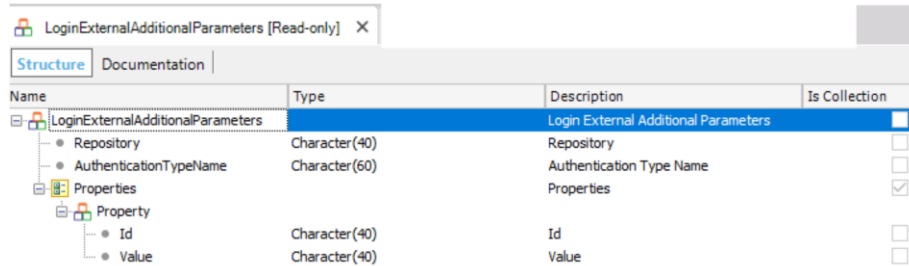
GeneXus

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	<input checked="" type="checkbox"/>
Property			
Id	Character(40)	Id	<input type="checkbox"/>
Value	Character(40)	Value	<input type="checkbox"/>

GeneXus

In the case of the SD login method, the LoginExternalAdditionalParameters allows to establish the Repository GUID to which to connect to. It's useful when there is more than one [GAM Repository Connection](#) in the connection.gam file on the server. Given a value to the Repository property of the &LoginExternalAdditionalParameters parameter, you can establish the connection to use (any connection in the connection.gam which refers to this Repository GUID).

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

If you have a multi-company SD app, with GAM; where you have defined more than one repository (one per company), to establish which repository to connect to, there is a new way to do it.

Hasta ahora se hacía llamando a un procedure que ejecutara en el server, y usara el método SetConnection (dado un GAM Connection name que estuviera definido en el GAM y en el connection.gam, se establecía la conexión a ese repositorio).

Sac

43580

El connection.gam en el server debe contener una entrada con ese Repository GUID, y debe tener más de una entrada en total, para que se reconozca ese parámetro.

Más información:

<https://wiki.genexus.com/commwiki/servlet/wiki?27708,HowTo%3A+Multiple+Repositories+Scenario+in+an+SD+application>,

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

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)

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

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

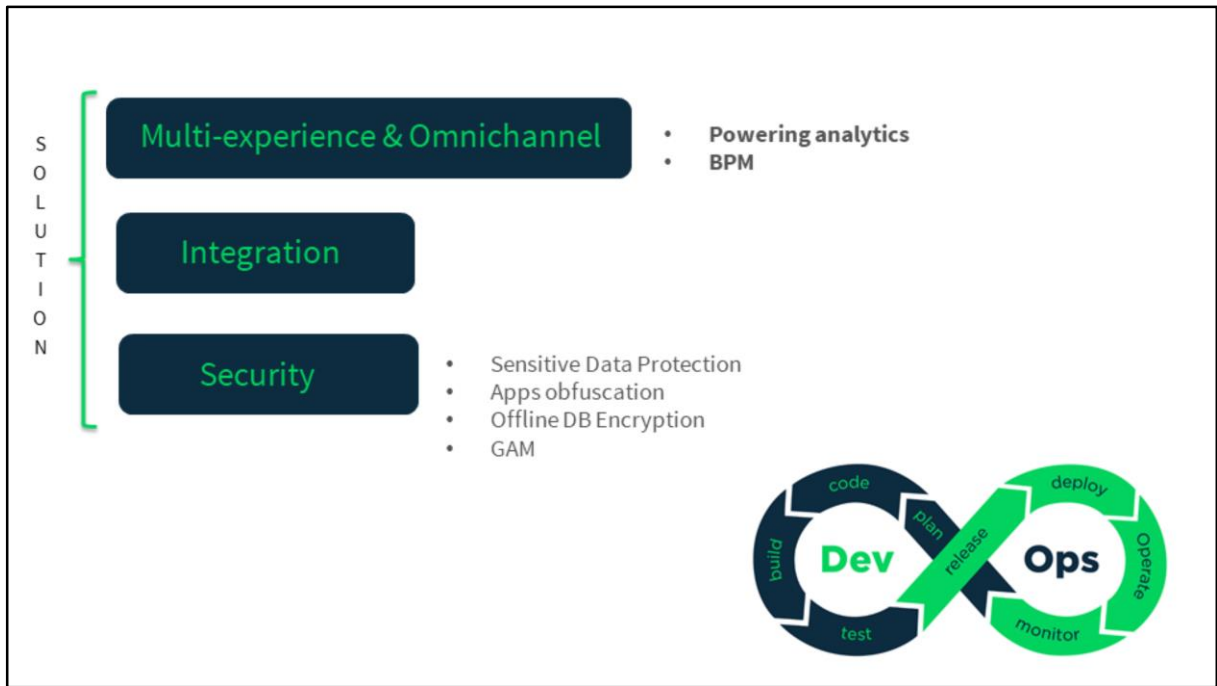
SAC 43466

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>,

GeneXus

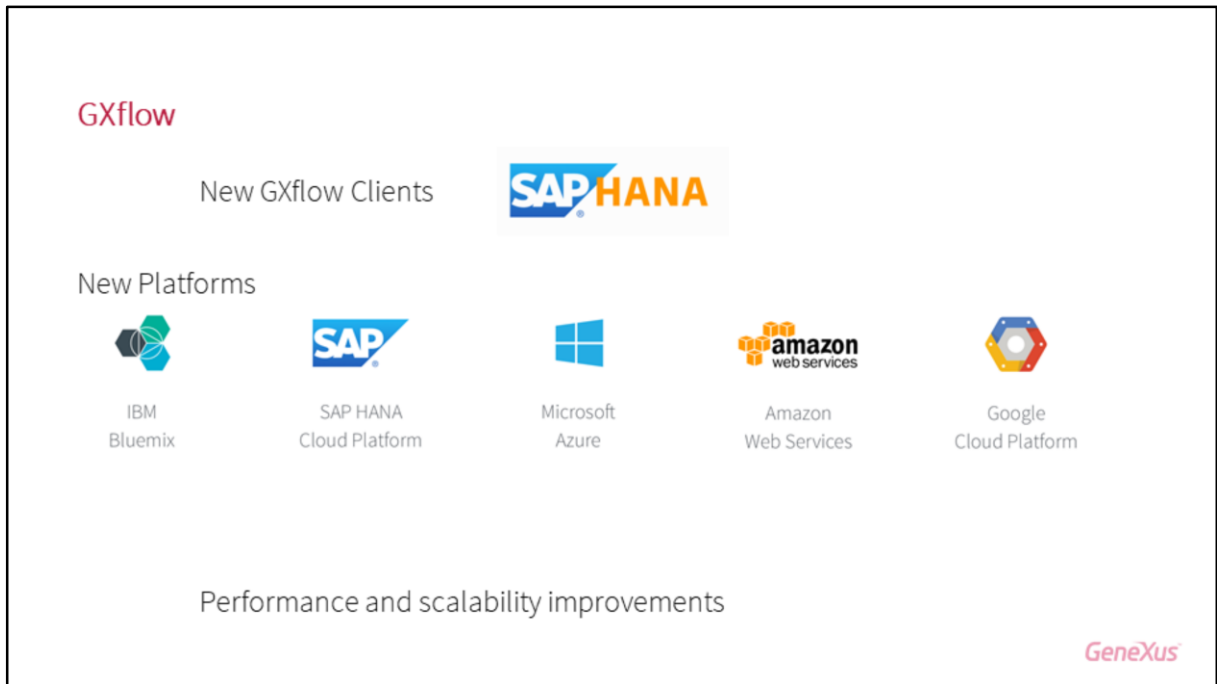
If you have a multi-company SD app, with GAM; where you have defined more than one repository (one per company), to establish which repository to connect to, there is a new way to do it.



Hemos visto aspectos de seguridad. Ahora veremos cómo dar más poder a la analítica de los datos, y también veremos los cambios en BPM.

BPM

GeneXus™ 16



GXflow adds SAP Hana Platforms

Gxflow soporta el cliente Java para SAP HANA Database.

GXflow on Paas

GXflow puede correr en cualquier Plataforma como servicio ([PaaS](#)) como ser: Amazon Beanstalk, Microsoft Azure, IBM Bluemix or SAP Cloud Platform.

Performance and scalability improvements

GXflow es una herramienta desarrollada con GX, entonces se aprovecharon varias de las funcionalidad de la versión 16 para mejorar la herramienta.

En el caso de los For Each internos se comenzó a utilizar el mecanismo de paginado en el dbms. Para ello en la clausula For each tiene las propiedades SKIP y COUNT.

También se implementaron mecanismos de Caching, se utiliza en Caching de tablas, de parámetros y de sesiones.

GXflow license

PROTECTION METHOD

Protection Server ▾

HOST

USER

PASSWORD

☐ OFFLINE AUTHORIZATION

SAVE

LICENSES



Request License



Enter License

Product	Version	Status	Type	Start Date	End Date	Copies	Unassig...
GXFLOW	16	AUTHORIZED	NOMINATED	09/18/18	09/17/18	100	100
GXFLOW CORPORATE	16	NOT AUTHORIZED	CORPORATE				

geneXus

Hasta ahora para el producto de Gxflow se tenían dos tipos de licencias que eran nominadas y corporativas, pero esto no se veía reflejado en el license manager. A partir de ahora se muestran como dos productos diferentes.

Además de esto se agregó la protección Nativa a nivel de la base de datos. Esto facilita la instalación de Gxflow en servidores PAAS y en ambientes Linux.

La otra protección sigue funcionando igual que funcionaba hasta ahora. La única diferencia es que todo el manejo de las licencias se puede hacer desde la interfaz web y no hay que ir al protection server para hacerlo.



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'  
.....
```



A partir de esta versión GXflow implementa log de todas sus funcionalidades. Para esto hace uso de la nueva funcionalidad de GX Log external object

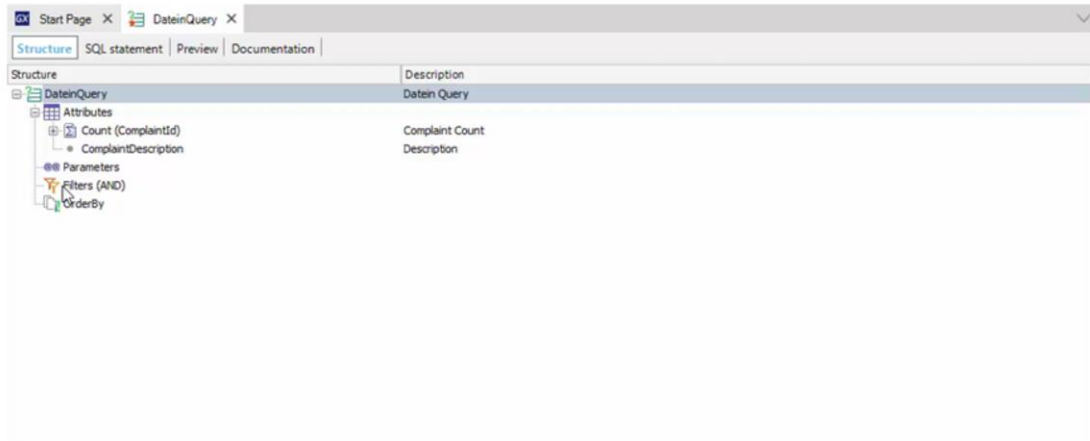
(<https://wiki.genexus.com/commwiki/servlet/wiki?37872,Log+external+object,>)

Esto aplica si se habilita la propiedad “Log level property” a “Debug” o “All”, esto ayuda a determinar problemas cuando se usan los tipos de datos de Gxflow.

Analytics: Reporting and Dashboards

GeneXus™ 16

&Today en el Query

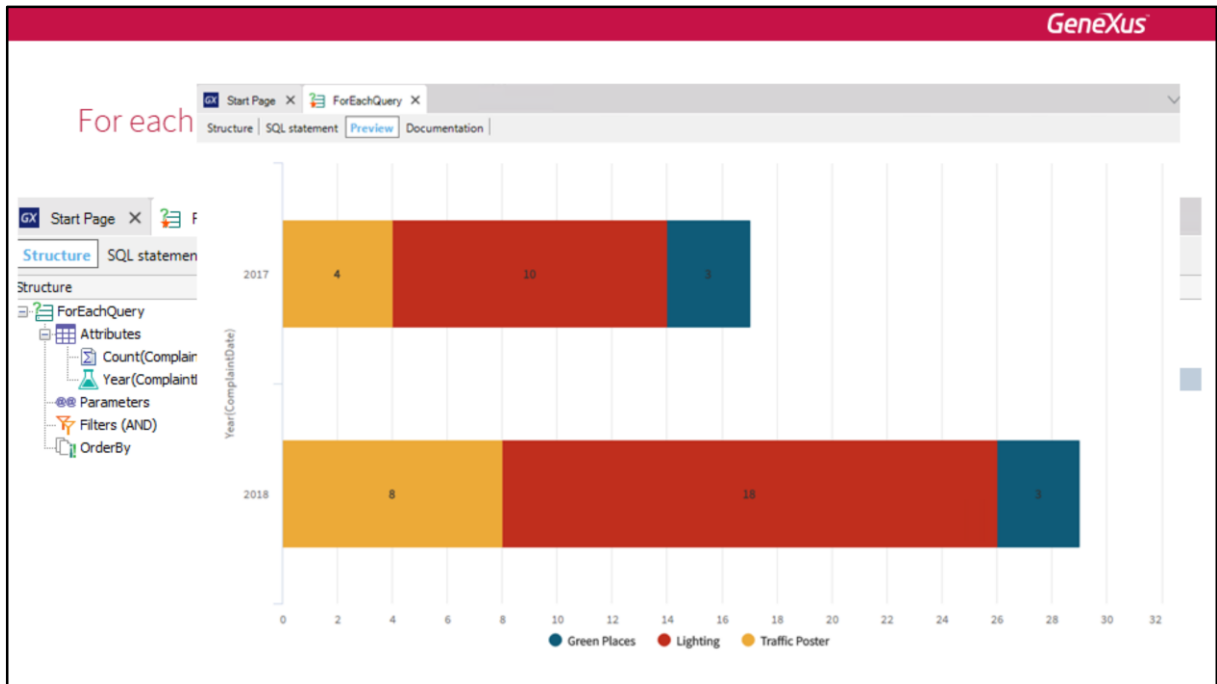


Al objeto query se le ha agregado más potencia, como se puede usar la variable &today en las condiciones de los objetos query.

Acá tenemos una consulta que nos muestra la cantidad de quejas (o reclamos) que fueron ingresadas de cada tipo y podemos mostrar solo aquellas que fueron ingresadas hasta el día de hoy. Para eso colocamos en las condiciones de la query “ComplaintDate <= &Today”

La estética del Query Viewer fue cambiada como se puede ver aquí en el preview. Además de haberse agregado nuevas gráficas para darle más potencia a las consultas. También se puede manejar el Width y Height en el control Query Viewer en términos de %.

Y se le pueden asignar al QueryViewer clases del objeto Theme. De esta forma la personalización del mismo queda toda dentro del tema de GeneXus.

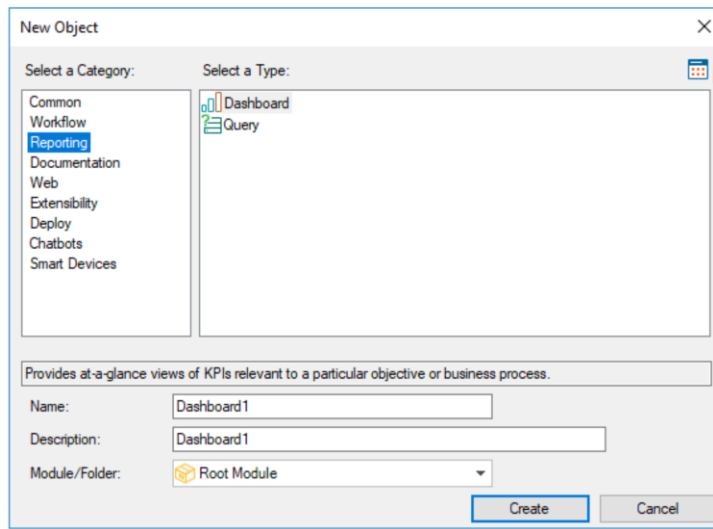


Se permite la cláusula For Each para el objeto Query. Esto hace que un indicador se pueda dividir en varios por una dimensión seleccionada.

En este caso estamos contando la cantidad de Quejas (complaints) que ingresaron los usuarios por tipo de queja para cada año.

Entonces en este caso quedaría así: `Count(ComplaintId) for each ComplaintReasonName`

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 with a blue selection bar), 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 with 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 'Dashboard1', 'Description:' with the value 'Dashboard1', and 'Module/Folder:' with a dropdown menu showing 'Root Module'. At the bottom right, there are two buttons: 'Create' and 'Cancel'.

Un Dashboard es una representación gráfica donde se presentan los principales indicadores del negocio, de una manera ordenada, clara y resumida.

Desde hace varios años tenemos en GeneXus el objeto Query que nos ayuda en esta tarea, y usando este objeto podemos construir en un WebPanel un dashboard, pero esta tarea implica cierto grado de programación (por ejemplo para enganchar los filtros con las consultas y disparar la actualización cuando se modifica un filtro).

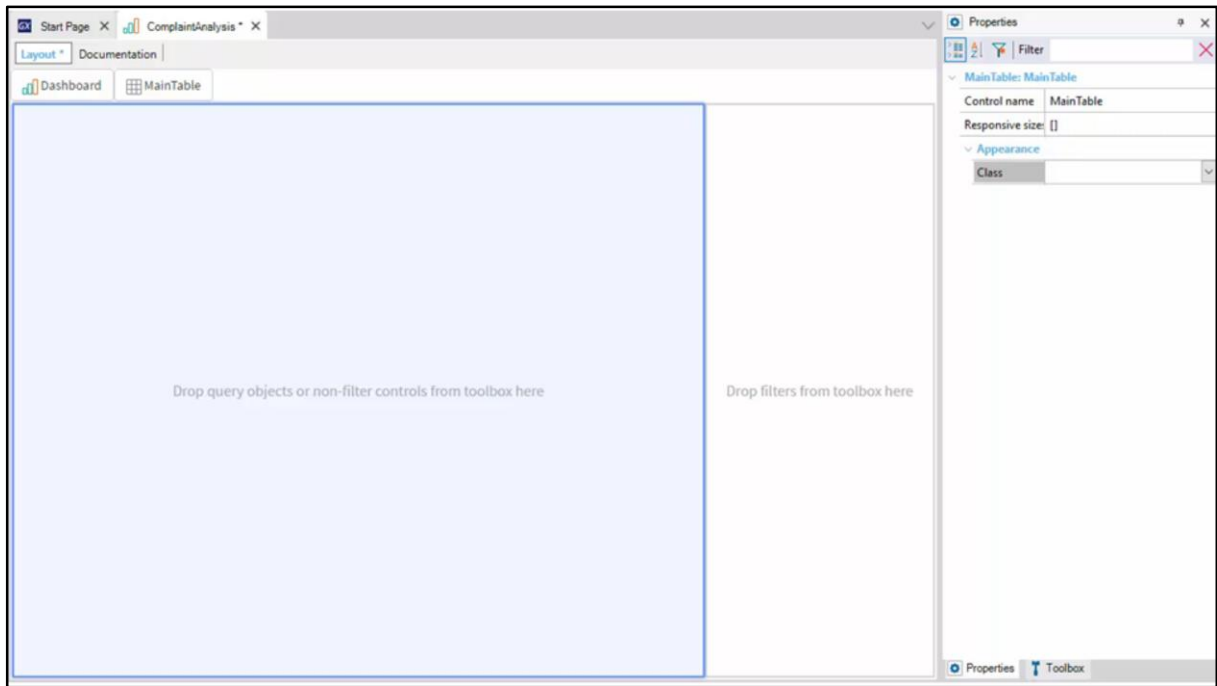
Lo nuevo del objeto Dashboard es que ahora podemos hacer lo mismo sin programar. Es decir lo podemos hacer de forma declarativa.

Dashboard Object

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

Las principales características son:

- No requiere programación. Todo se hace arrastrando componentes a la pantalla y configura propiedades de los mismos.
- Los tipos de componentes que se pueden usar son:
 - **Query**, para mostrar información
 - **Filtro**, para seleccionar un valor (o valores) y de esta manera filtrar la información que se muestra en las Queries
 - **Imagen**, para mostrar una imagen
 - **TextBlock** para mostrar un texto.
- Tiene edición “en vivo”. Mientras se va construyendo el Dashboard ya vemos como va a quedar finalmente. Si se tiene la conexión a la base de datos ya se ven los datos reales. Además todos los cambios de estética que se hacen se van viendo en el momento.
- Como el Dashboard es una definición de la presentación de la información. No es un objeto ejecutable en si mismo. Para poder ejecutarlo necesitamos de un User Control, y ese User Control es el DashboardViewer.



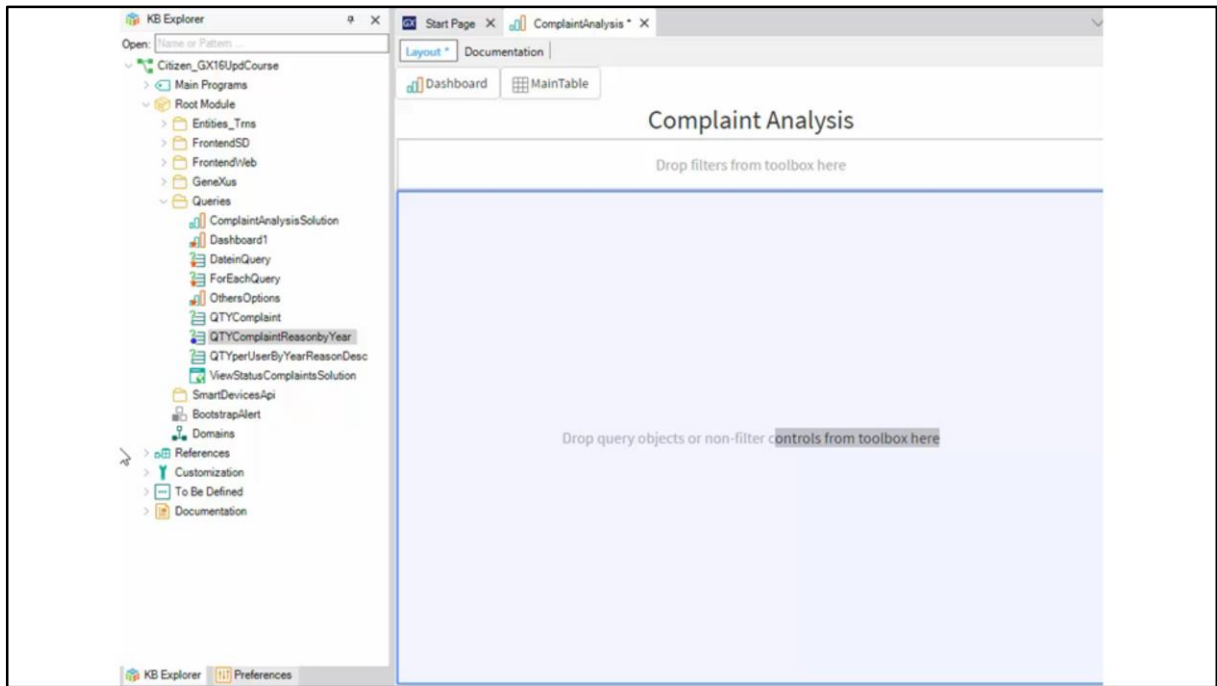
Cuando creamos un objeto Dashboard lo primero que se ven son dos áreas, el área donde van las consultas y el área de los filtros.

Estas áreas se pueden ubicar de diferentes formas.

La posición de los filtros puede configurarse simplemente modificando una propiedad (es posible disponerlos a la derecha, izquierda o encima de los demás componentes)

El editor del Dashboard viene con varios tipos de layout predeterminados (2 o 3 columnas, con o sin encabezado y pie)

Es posible configurar que la información del dashboard se actualice automáticamente cada cierta cantidad de segundos



Una vez que tenemos el objeto dashboard creado podemos colocar las queries que ya tenemos en nuestra kb creadas para ir completándolo.

Si colocamos queries que reciben parámetros automáticamente se crean los parámetros que estas consultas van a necesitar en el área de parámetros.

Como los parámetros de las consultas eran sobre atributos de la kb, GX automáticamente detectó que podían cargarse los valores a ser seleccionados. Entonces los creo como un Dynamic combo box.

Los tipos de controles disponibles dependen del tipo de datos del filtro y también si el filtro es un valor atómico, un rango o una collection. Los tipos de controles incluyen: Edit box, Combo box, Radio Button, Check boxes, Slider, Drop-down list y Switch

Start PageComplaintAnalysis *QTYperUserByYearReasonDesc

Layout *Documentation

DashboardFiltersFilter1

Complaint Analysis

complaintreasonname

Lighting

complaintdescription

Car parked in front of a garage

year

0

Drop filters from toolbox here

Drop filters here

User NameCount (Complaint Id)

Drop query objects or non-filter controls from toolbox here

Properties

Filter

Filter: Filter1

Control name

Filter1

Type

Value

Data type

Character

Name

complaintreasonname

Value

Lighting

Visible

True

Caption

complaintreasonname

Picture

Frame

Visible

False

Control

Type

Combo box

Dynamic

True

Item values

ComplaintReasonName

Item description

ComplaintReasonName

Sort description

True

Empty item

False

Conditions

Appearance

Caption class

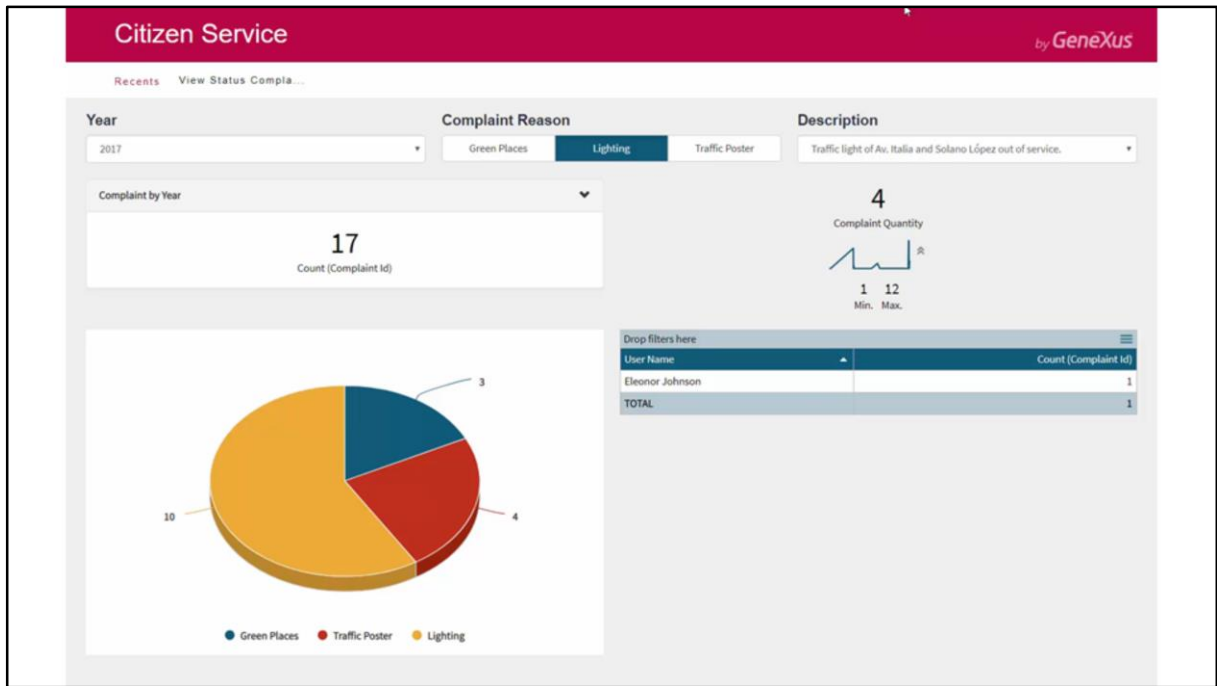
Value class

Container class

Cell information

Cell class

PropertiesToolbox



Este objeto sigue evolucionando

- Pasaje de parámetros
- Más eventos, por ejemplo llamar a un objeto GeneXus al hacer clic sobre un valor
- Posibilidad de organizar los componentes del Dashboard en tabs
- Posicionamiento absoluto de los componentes

GeneXus™
The power of doing