

Curso práctico de actualización a

GeneXus™ 16

PARTE 4

Abril 2020

Copyright © GeneXus S.A. 1988-2020.

All rights reserved. This document may not be reproduced by any means without the express permission of GeneXus S.A. The information contained herein is intended for personal use only.

Registered Trademarks:

GeneXus is trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners.

CONTENIDO

CONTENIDO.....	2
OBJETIVO.....	3
KB DE PARTIDA	3
ENTIDADES QUE UTILIZAREMOS.....	4
OBJETO DASHBOARD.....	4
Solución.....	5
KB SOLUCIÓN	10

OBJETIVO

En este práctico trataremos sobre el nuevo objeto Dashboard para integrar consultas de manera vistosa en sus aplicaciones web.

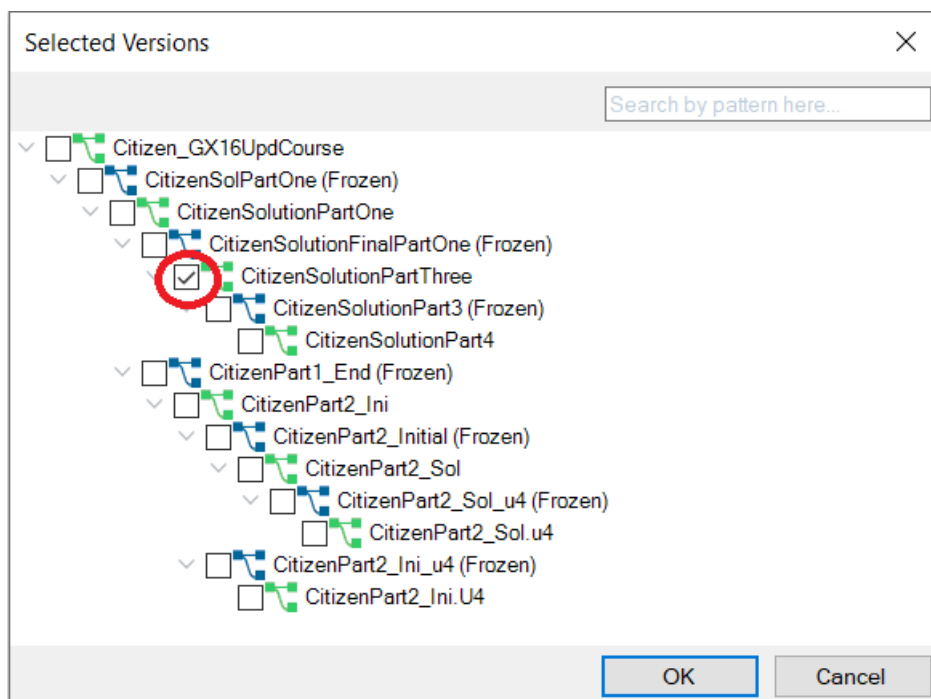
La letra de este práctico se ha actualizado de acuerdo al upgrade 8 de GeneXus 16. Si lo realiza con una versión posterior, considere que puede haber cambios entre lo que aquí se le muestra y lo que usted ve.

Continuaremos utilizando la KB Citizen en el estado resultante del práctico anterior. Si por algún motivo no conserva esa KB, entonces en el punto siguiente se le explica cómo obtener una similar.

Recuerde que la aplicación es una simplificación de una app para la municipalidad de una ciudad, que ofrece un frontend (Web/SD/Conversational) para que los ciudadanos mediante su identificador de usuario puedan realizar reclamos (por ejemplo por árboles caídos, semáforos que no funcionan, coches mal estacionados, etc.), puedan realizar trámites (por ejemplo para obtener licencia de conducir, refinanciar una deuda con el municipio, instalar elementos de publicidad, etc), reservando un turno para ser atendidos por el personal del municipio. En el frontend se les muestran también las diversas actividades culturales que ofrece la ciudad. Y también se cuenta con un backoffice web para que ciertos funcionarios de la municipalidad manejen los datos y vean estadísticas. A este backoffice es que agregaremos el dashboard.

KB DE PARTIDA

Se partirá de la versión de la KB con la que usted trabajó en el práctico anterior. Si por algún motivo la perdió o no la encuentra, créese otra desde <http://samples.genexusserver.com/v16>, (KB de nombre Citizen_GX16Course) eligiendo la versión de nombre **CitizenSolutionPartThree**.



Nota: si desea que funcionen los servicios ODATA deberá volver a indicar para el Data Store “ServiceDS1 (Service)” la propiedad: Server name = [https://services.odata.org/V4/\(S\(40gwcqlhjmuyfayfnplo00\)\)/TripPinServiceRW/](https://services.odata.org/V4/(S(40gwcqlhjmuyfayfnplo00))/TripPinServiceRW/)

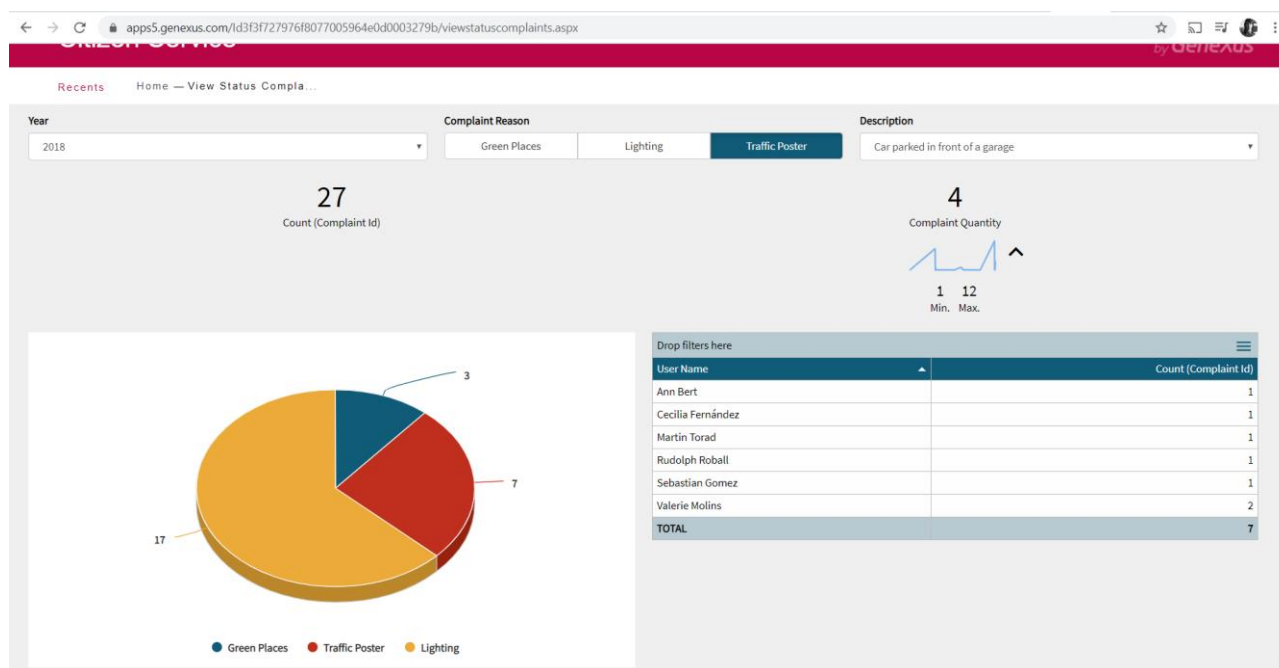
Realice un rebuild all antes de empezar.

ENTIDADES QUE UTILIZAREMOS

Bajo el folder Entities_Trns se encuentran las entidades que modelan la realidad de la app. En este práctico nos concentraremos en los reclamos (complaints) de los usuarios de acuerdo a las categorías de reclamos (razones de los reclamos, ComplaintReason).

OBJETO DASHBOARD

Queremos que el backoffice ofrezca una pantalla como la que sigue, donde se puedan consultar los reclamos (complaints) realizados por los usuarios y filtrarlos por “Year”, “Complaint Reason” y “Complaint Description”:



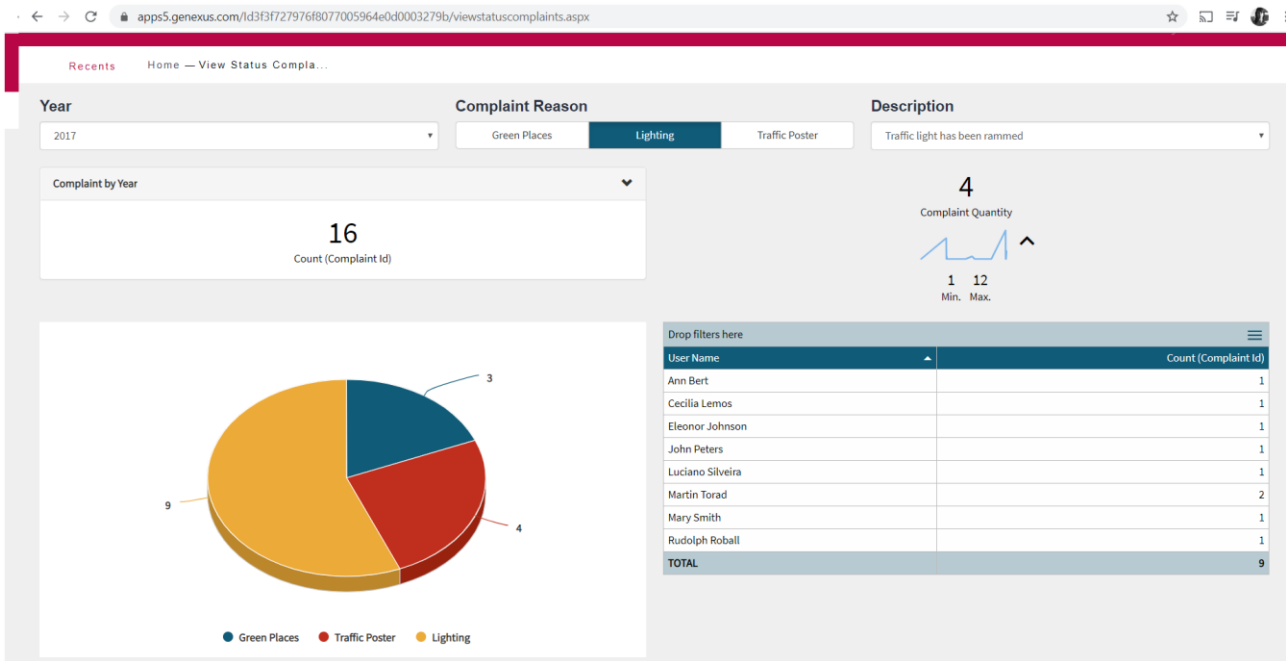
¿Cómo lo implementa? Una vez hecho, pruébelo en ejecución.

SOLUCIÓN

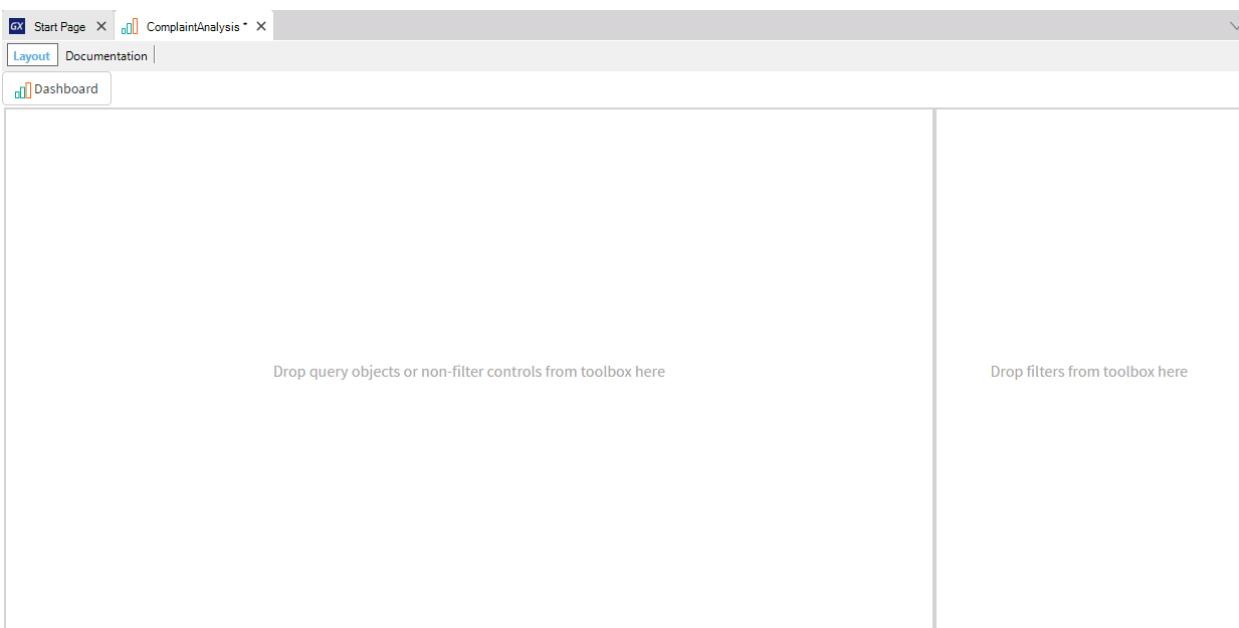
Para ello nos vamos a crear un objeto de tipo Dashboard (de la categoría Reporting) de nombre “ComplaintAnalysis”.

Para utilizar este dashboard vamos a utilizar las consultas que ya tenemos creadas que se encuentran en el folder “Queries”.

La idea es tener un dashboard como el que se muestra en la siguiente figura.



Al inicio el objeto Dashboard cuenta con dos áreas, la de “Queries” y la de “Filters”



La idea es colocar las consultas haciendo Drag and Drop en el área de consultas, en la siguiente ubicación y con el siguiente tipo:

- Ángulo superior izquierdo – QTYComplaintReasonbyYear de tipo Card.
- Ángulo superior derecho – QTYComplaint de tipo Card.
- Ángulo inferior izquierdo – QTYComplaintReasonbyYear de tipo Chart/Pie
- Ángulo inferior derecho – QTYperUserByYearReasonDesc de tipo Pivot table.

Al colocar las consultas GeneXus automáticamente ya definió los filtros que las consultas tenían. Ahora se deberían cambiar los caption de los filtros y modificar el tipo para que queden como la imagen inicial.

- “Year” de Tipo Combo Box estático, con los valores 2017 y 2018
- “Complaint Reason” de Tipo Radio Button dinámico, donde los valores se toman del atributo “ComplaintReasonName”
- “Description” de Tipo Combo Box dinámico, donde los valores se toma del atributo “ComplaintDescription” y se le agrega la condición “ComplaintReasonName = &ComplaintReasonName”; de esta forma van a quedar relacionados los Filtros “Complaint Reason” y “Description”

Una vez terminada la configuración de los filtros ya se pueden modificar ya que se tiene un live editing del Dashobard.

Tip: Para colocar los filtros en la parte superior del objeto Dashboard, hay que posicionarse en la sección Layout, ahí se habilitan las propiedades y podemos cambiar en “Filters position” la posición a Top, en lugar de Right.

The screenshot displays the GeneXus IDE interface for configuring a dashboard. The main workspace shows a dashboard with several components:

- Filters:** Year (2018), Complaint Reason (Traffic P...), and Description (Car parked in front of a garage).
- KPI Card:** Count (Complaint Id) with a value of 25.
- Chart:** Complaint Quantity, showing a range from 1 to 12.
- Pie Chart:** Distribution of complaint reasons: Green Places (6), Lighting (16), and Traffic Poster (3).
- Table:** A table with columns 'User Name' and 'Count (Complaint Id)'. The data is as follows:

User Name	Count (Complaint Id)
Ann Bert	1
Cecilia Fernández	1
Martin Torad	1
Sebastian Gomez	1
Valerie Molins	2
TOTAL	6

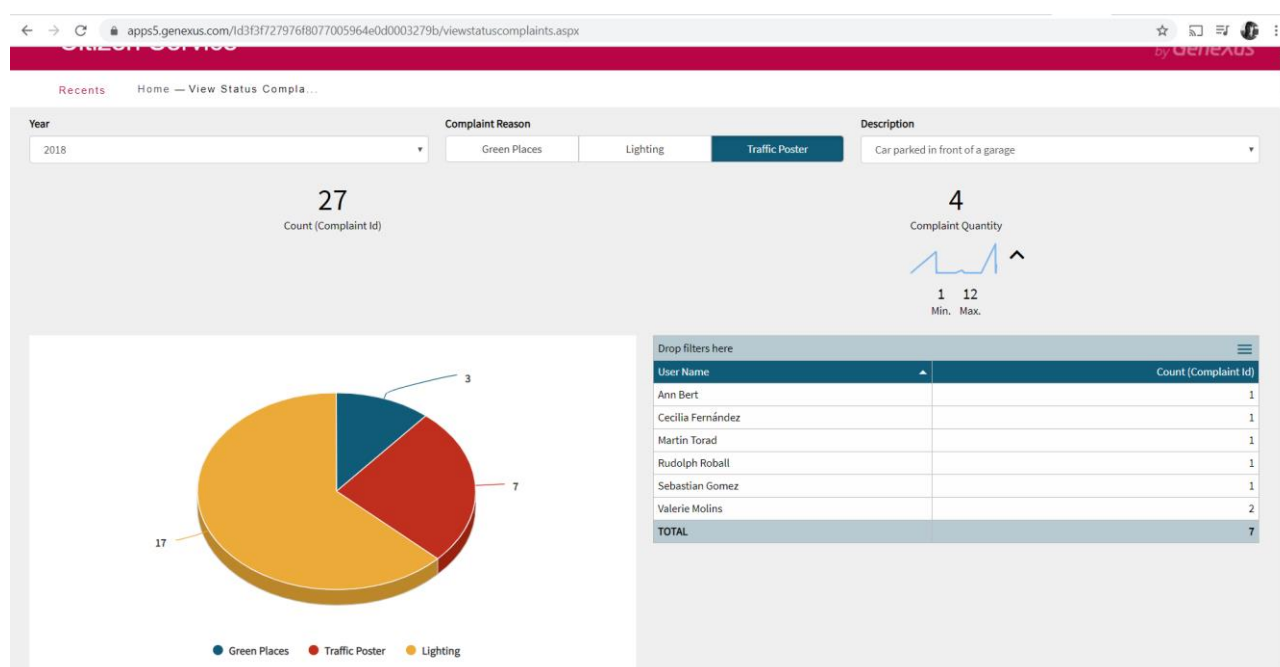
The 'Layout' tab is selected, and the 'Properties' window on the right shows the 'Filters position' property set to 'Top'.

Ahora es momento de agregar el Dashboard a nuestra aplicación. Para ello vamos a crear un Webpanel llamado "ViewStatusComplaints". En ese webpanel vamos a agregar un control de tipo DashboardViewer. Al control le vamos a asociar el objeto dashboard creado anteriormente.

Para finalizar el ejercicio y poder incorporar el webpanel creado al backoffice, podemos agregarlo al objeto "ListPrograms", agregando las siguientes líneas:

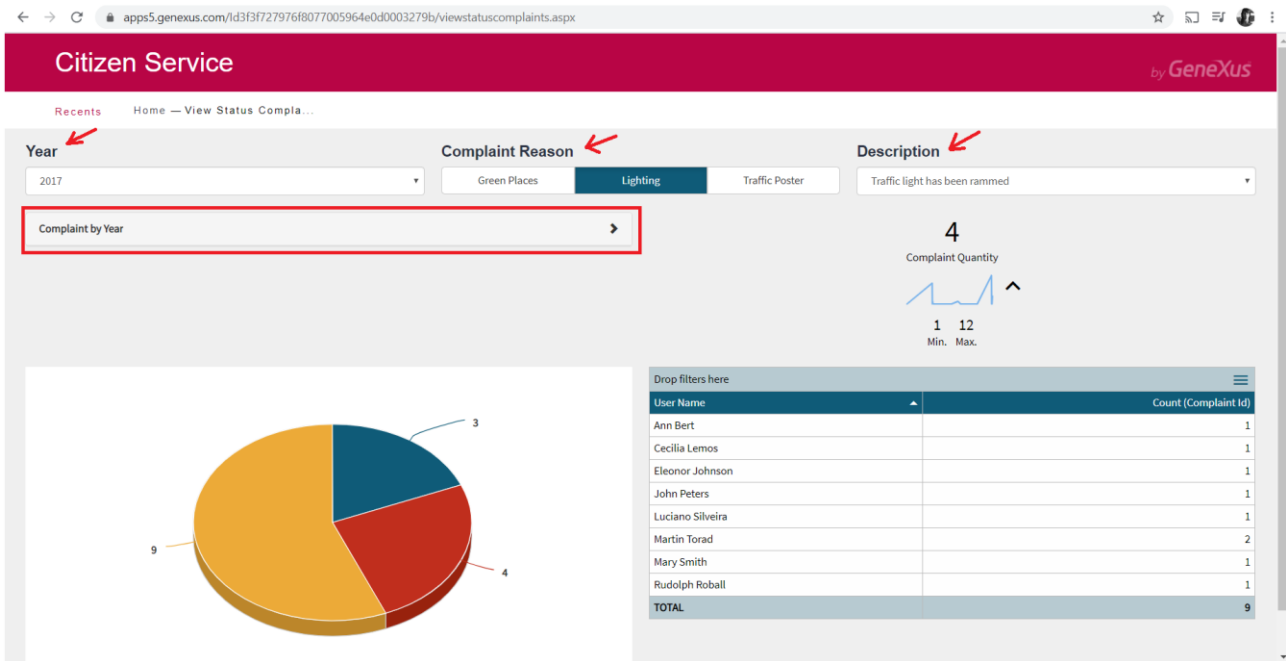
```
&name = !"ViewStatusComplaints"  
&description = "View Status Complaints"  
&link = ViewStatusComplaints.Link()  
  
Do 'AddProgram'
```

De esta forma al ejecutar el Home de nuestro backoffice (Folder GeneXus/Web) vamos a encontrar la invocación a nuestro "ViewStatusComplaints" y podremos ver nuestro Dashboard en ejecución:



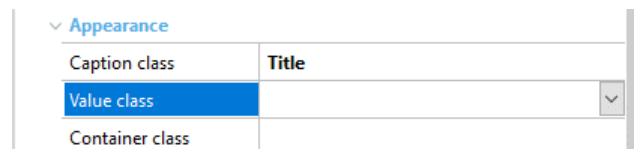
La idea ahora es modificar los filtros y ver cómo se comportan las diferentes consultas.

En esta nueva etapa queremos cambiar un poco la estética de las consultas, el resultado al que queremos llegar es el que se muestra a continuación.



Las modificaciones que se quieren realizar son cambiar el tamaño del título de los filtros y también que la consulta que está como Card se muestre colapsada.

Tip: Para cambiarle la clase asociada al Título, ir a las propiedades en Appearance, Caption class y seleccionar la clase Title.



Para modificar la consulta `QTYComplaintReasonbyYear`, se tiene que cambiar Frame a Visible, colocarle un título y habilitar a que se pueda colapsar.

Object: Object3	
Control name	Object3
Object	QTYComplaintReasonbyYear
Frame	
Visible	True
Title	Complaint by Year
Allow collapsing	True
Collapsed	False
Output	
Type	Card
Show data as	Values
Orientation	Horizontal
Include trend	False
Include sparkline	False

Properties | Toolbox

Luego de todos esos cambios salvar el dashboard y ejecutar nuevamente la aplicación, ya que estos cambios no se ven dentro de GeneXus sino en runtime.

KB SOLUCIÓN

Podrá descargar de GeneXus Server la KB solución de este práctico para comparar resultados. Es la versión de la KB de nombre CitizenSolutionPart4.

Corrobore que en el Data Store “ServiceDS1 (Service)” la propiedad de conexión Server name haya quedado así: [https://services.odata.org/V4/\(S\(40gwjclhjmuyfayfnplo0o\)\)/TripPinServiceRW/](https://services.odata.org/V4/(S(40gwjclhjmuyfayfnplo0o))/TripPinServiceRW/)

Y no vacía. Si está vacía, copie el valor anterior.