

# Práctico de Dashboards

**GeneXus™ 16**

Setiembre 2019

*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.....	3
OBJETO DASHBOARD.....	3
Solución.....	4

## OBJETIVO

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

Utilizaremos la KB Citizen, que es una simplificación de una aplicación para la municipalidad de una ciudad. Dentro de esta Kb vamos a encontrar los reclamos que pueden hacer los ciudadanos (por ejemplo por árboles caídos, semáforos que no funcionan, coches mal estacionados, etc.), la posibilidad de realizar trámites (por ejemplo para obtener licencia de conducir, refinanciar una deuda con el municipio, instalar elementos de publicidad, etc). Dentro de este backend web que manejan los funcionarios de la municipalidad, le vamos a agregar un objeto Dashboard para que puedan ver las estadísticas del sistema.

## KB DE PARTIDA

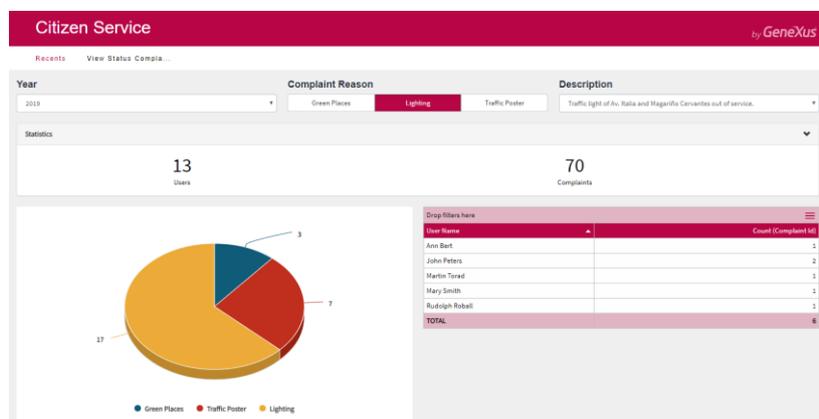
Se partirá de la KB que se encuentra en GXServer: <http://samples.genexusserver.com/v16/> con el nombre Citizen\_dashboards:

## ENTIDADES QUE UTILIZAREMOS

Bajo el folder Entities\_Trns se encuentran las entidades que modelan la realidad de la aplicación. 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 la aplicación 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”:



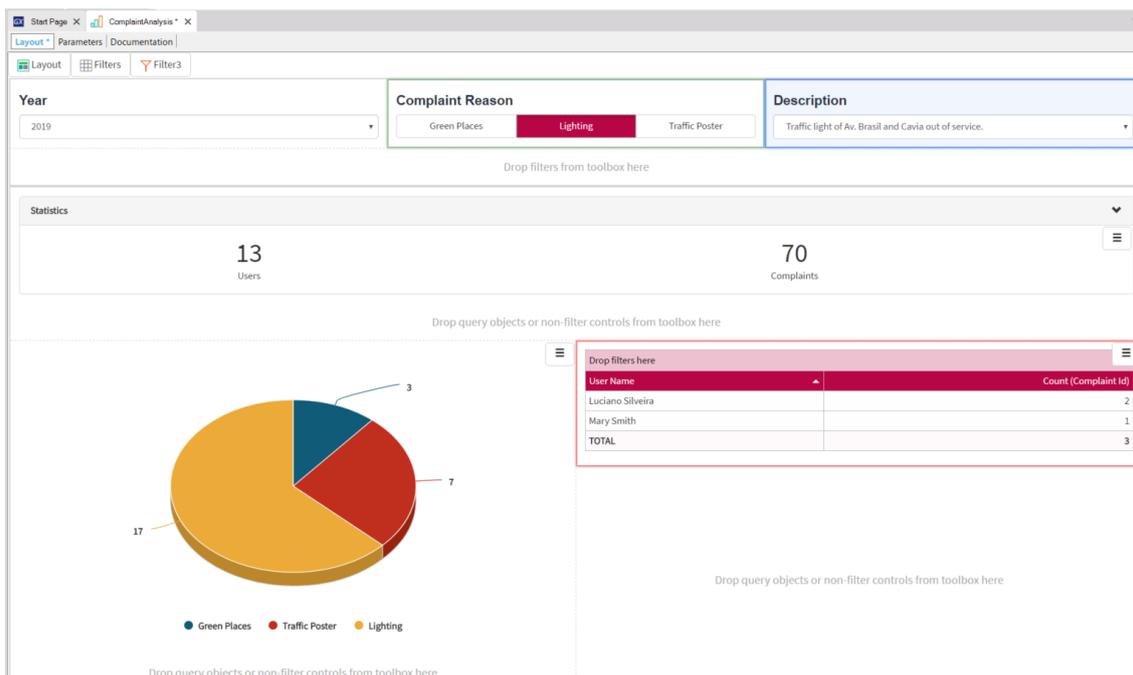
¿Cómo lo implementa? Una vez hecho, pruébelo en ejecución.

## SOLUCIÓN

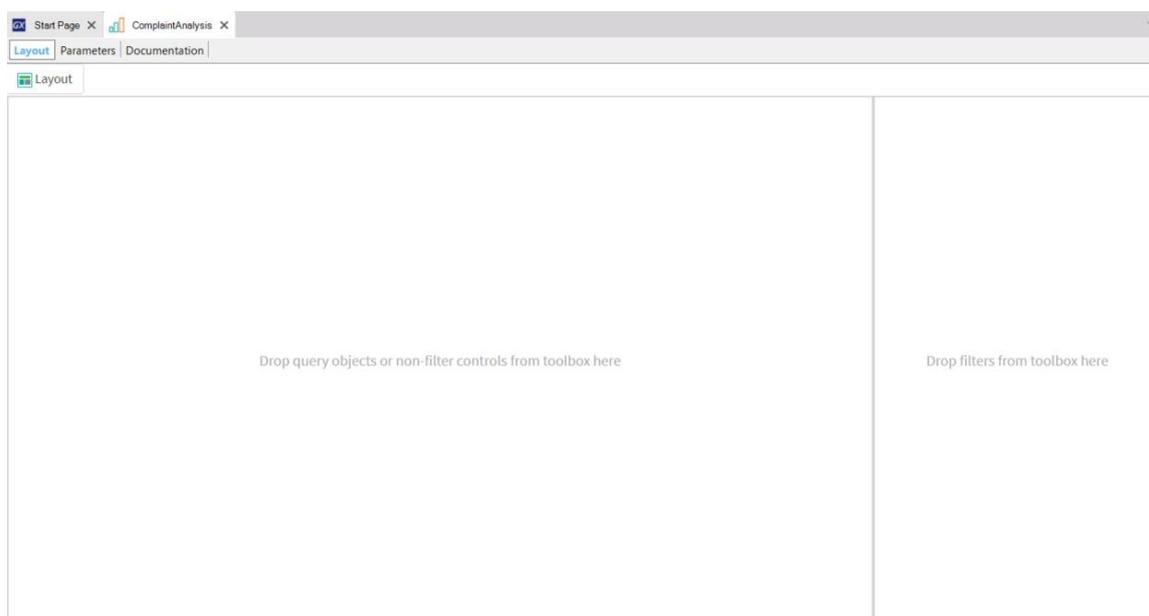
Para ello nos vamos a crear un objeto de tipo Dashboard (dentro de la categoría Reporting) de nombre “ComplaintAnalysis”.

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:

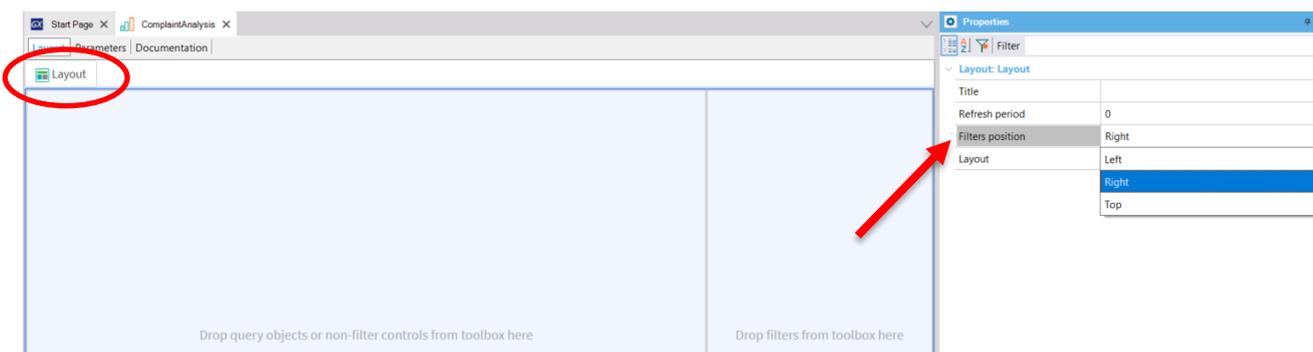
- Arriba – QTUsers de tipo Card.
- Ángulo inferior izquierdo – QTYComplaintReasonbyYear de tipo Chart.
- Ángulo inferior derecho – QTYperUserByYearReasonDesc de tipo Pivot table.

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.

- “Years” de Tipo Combo Box estático, con los valores 2017, 2018 y 2019
- “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.



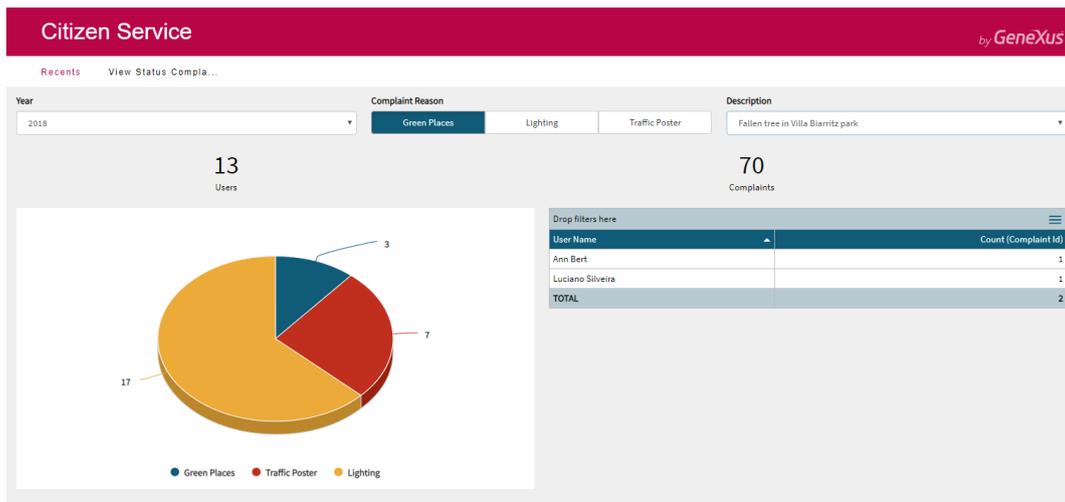
Ahora es momento de utilizar el Dashboard creado en un webpanel. Para ello abrimos el Webpanel “ViewStatusComplaints”. En ese webpanel vamos a agregar un control de tipo DashboardViewer y para asociar el Dashboard creado anteriormente vamos a descomentar el evento Refresh, que contiene lo siguiente:

```

Start Page X ComplaintAnalysis X ViewStatusComplaints X
Web Form | Rules | Events | Conditions | Variables | Help | Documentation
nts
1 Event Refresh
2   DashboardViewer1.Object = ComplaintAnalysis()
3 -Endevent
4

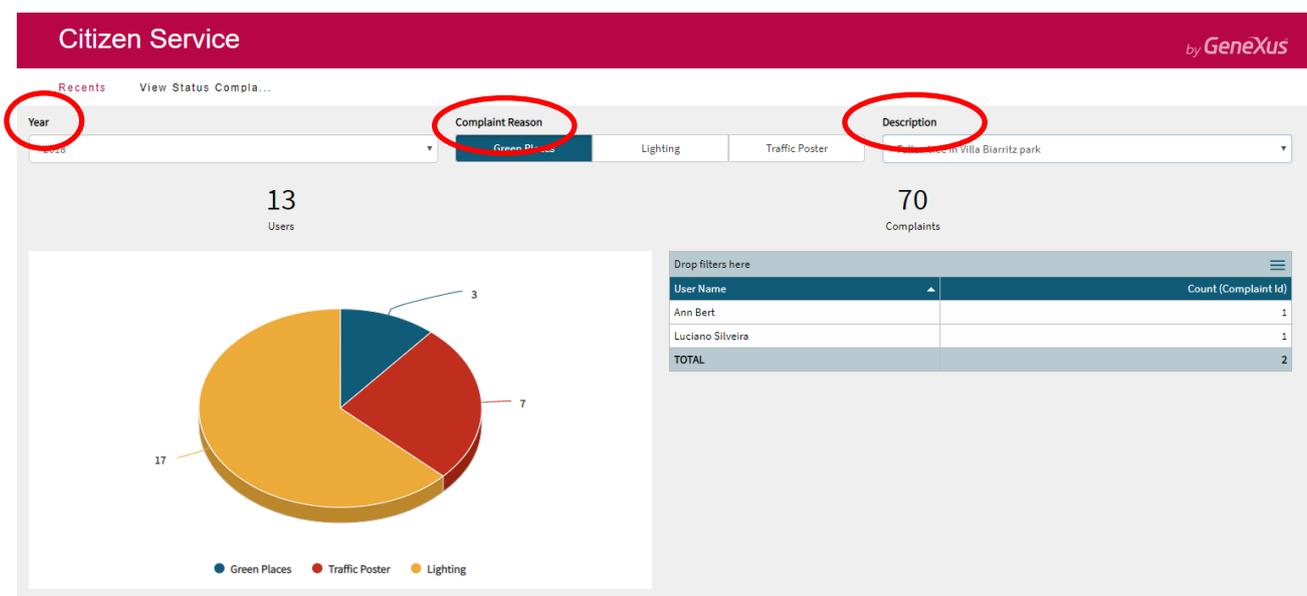
```

Ejecutar el Web Panel `ViewStatusComplaints` para ver todo lo realizado hasta ahora.



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.

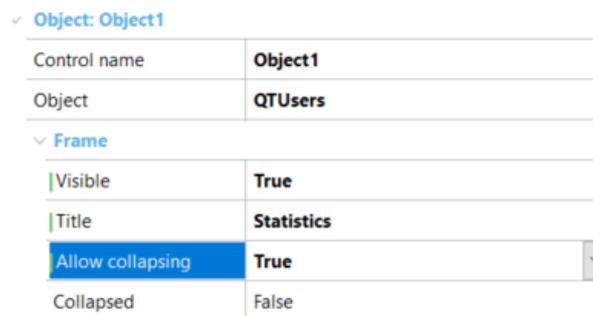


Las modificaciones que se quieren realizar son cambiar el tamaño del título de los filtros y también que la consulta que está como Card se muestre colapsada.

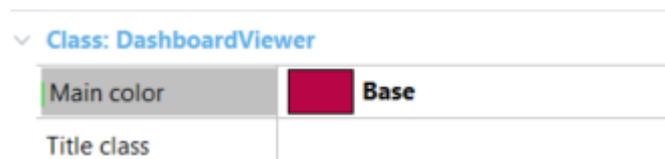
**Tip:** Para cambiarle la clase asociada al Título, seleccionar los tres filtros a la vez. Ir a las propiedades en **Appearance**, **Caption class** y seleccionar la clase **Title**.



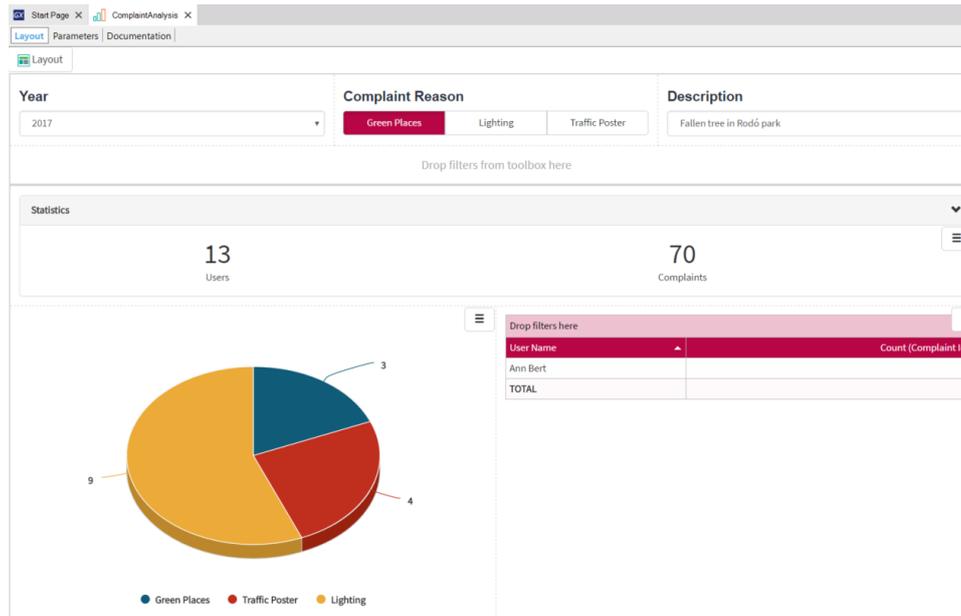
Para modificar la consulta **QTUsers**, se tiene que cambiar Frame a Visible, colocarle un título **Statistics** y habilitar a que se pueda colapsar.



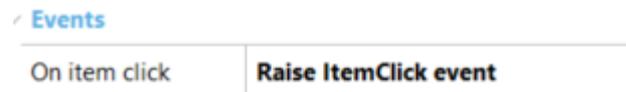
Para cambiar los colores del dashboard y que queden en las mismas tonalidades del tema **Carmin**. Ir al tema **Carmin**, seleccionar la clase **DashboardViewer** y cambiar la propiedad "Main color" por **Base**. Salvar el tema.



Luego de todos esos cambios salvar el dashboard, cerrar y abrirlo así se muestran todos los cambios. Debería quedar como la siguiente imagen



Ahora agreguémosle un poco más de complejidad al Dashboard, para ello vayamos al a Consulta QTUsers que es de tipo Card. Habilitemos el evento On ítem Click con el valor Raise ItemClick event



De esta forma al hacer clic va a ejecutar lo que se tenga definido en el evento. Este evento va a estar programado en el Webpanel “ViewStatusComplaints”, solo hay que des comentar el evento DashboardViewer1.ItemClick que ya está programado.

```

4
5
6 Event DashboardViewer1.ItemClick
7   If &ItemClickData.Object = "QTUsers"
8     Do Case
9       Case &ItemClickData.Element = "Users"
10        &Window.Autosize = false
11        &Window.Width = 400
12        &Window.Height = 600
13        &Window.Object = ViewUsers.Create()
14       Case &ItemClickData.Element = "Complaints"
15        &Window.Autosize = false
16        &Window.Width = 400
17        &Window.Height = 600
18        &Window.Object = ViewComplaints.Create()
19      endcase
20      &Window.Open()
21    endif
22  Endevent
23

```

Esto llame a dos Web Panels diferentes, uno para **Users** y otro para **Complaitns**. Vayamos al Web Panel **ViewUsers**, este tiene un **QueryviewerControl** asociado a la Query **TotalUsers**. En las propiedades de ese control tiene habilitado **Allow Selection = True**, lo cual va a permitir seleccionar un usuario y realizar la acción que esta programada en los eventos de este Web Panel

QueryViewer: UCWeb	
Control Name	QueryViewer1
Data bindings	
Object	TotalUsers
Axes	&Axes
Allow Change Axes Or	False
Parameters	&Parameters
Appearance	
Behavior	
Auto Refresh Group	
Disable Column Sort	False
Allow Selection	True
Remember Layout	True

En ejecución, probar seleccionar **Users**, ver que abre un panel con todos los usuarios y se puede seleccionar cada uno y ver datos particulares del mismo. Ejecutar el Web Panel **ViewStatusComplaints** para ver en ejecución todos estos cambios.

Ahora queremos que el Dashboard cuando abre, cargue por defecto el año actual 2019. Para eso vamos a definir en el Dashboard un parámetro en el área de parámetros



Además, en el filtro **Year** colocar en las propiedades en **Value =&Year**.

Luego vamos al Web Panel **ViewStatusComplaints**, y en la llamada al Dashboard le agregamos el valor 2019 como se muestra.

```

1 | Event Refresh
2 |     DashboardViewer1.Object = ComplaintAnalysis(2019)
3 | Endevent

```

Salvar y ejecutar nuevamente el web panel para ver los cambios.

**GeneXus™**  
[www.genexus.com](http://www.genexus.com)

MONTEVIDEO - URUGUAY  
CIUDAD DE MÉXICO - MÉXICO  
MIAMI - USA  
SÃO PAULO - BRASIL  
TOKYO - JAPAN

Av. Italia 6201- Edif. Los Pinos, P1  
Hegel N° 221, Piso 2, Polanco V Secc.  
7300 N Kendall Drive, Suite 470  
Rua Samuel Morse 120 Conj. 141  
2-27-3, Nishi-Gotanda  
Shinagawa-ku, Tokyo, 141-0031

(598) 2601 2082  
(52) 55 5255 4733  
(1) 201 603 2022  
(55) 11 4858 0300  
(81) 3 6303 9381  
(81) 3 6303 9980