

# Curso prático de atualização para

# GeneXus™ 16

## PARTE 4

Setembro 2018

*Copyright ©GeneXus S.A. 1988-2018.*

*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.*

## CONTEÚDO

CONTEÚDO.....	2
OBJETIVO.....	3
KB BASE .....	3
ENTIDADES QUE UTILIZAREMOS.....	4
OBJETO DASHBOARD.....	4
Solução .....	5
KB SOLUÇÃO .....	9

## OBJETIVO

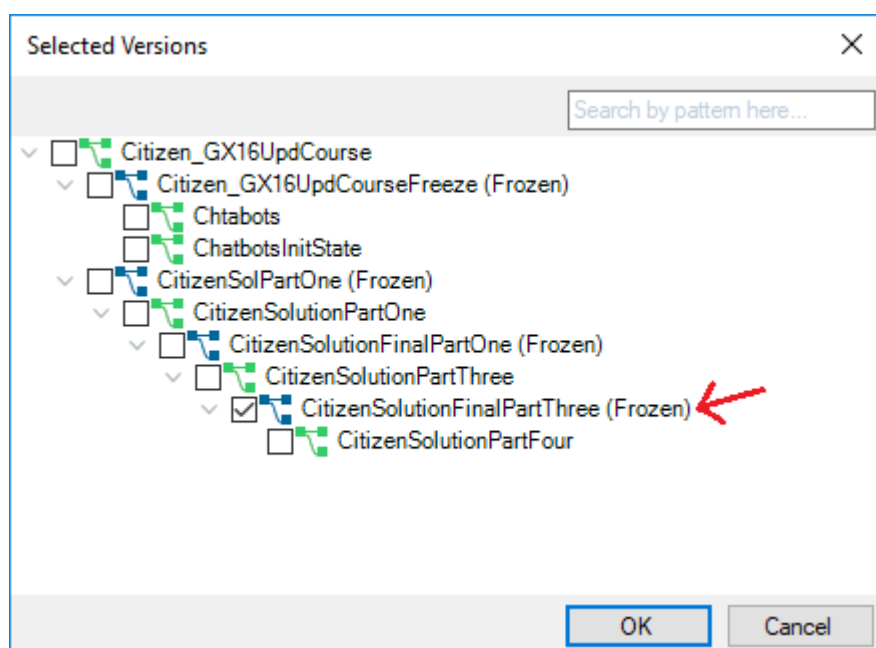
Neste prático, trataremos sobre o novo objeto Dashboard para integrar as consultas de maneira atraente em suas aplicações web.

Continuaremos utilizando a KB Citizen no estado resultante do prático anterior. Se por alguma razão não preservou essa KB, então no ponto seguinte é explicado como obter uma similar.

Lembre-se que a aplicação é uma simplificação de uma app para o município de uma cidade, que oferece um frontend (Web/SD/Conversational) para que os cidadãos, através de seu identificador de usuário, possam fazer reclamações (por exemplo, árvores caídas, semáforos que não funcionam, carros mal estacionados, etc.), possam realizar procedimentos (por exemplo, para obter uma carteira de motorista, refinanciar uma dívida com o município, instalar elementos publicitários, etc.), reservando um turno para ser atendido pela equipe do município. No frontend também são mostradas as diversas atividades culturais oferecidas pela cidade. E há também um backend web para que certos funcionários do município lidem com os dados e vejam estatísticas. A este backend é que vamos adicionar o dashboard.

## KB BASE

Será baseado na versão da KB com a qual você trabalhou no prático anterior. Se por algum motivo você a perdeu ou não a encontra, crie outra a partir de <http://samples.genexusserver.com/v16>, (KB chamada Citizen\_GX16Course) escolhendo a versão chamada **CitizenSolutionFinalPartThree**.



**Nota:** se você deseja que os serviços ODATA funcionem, deve voltar indicar para o Data Store “ServiceDS1 (Service)” a propriedade: Server name =

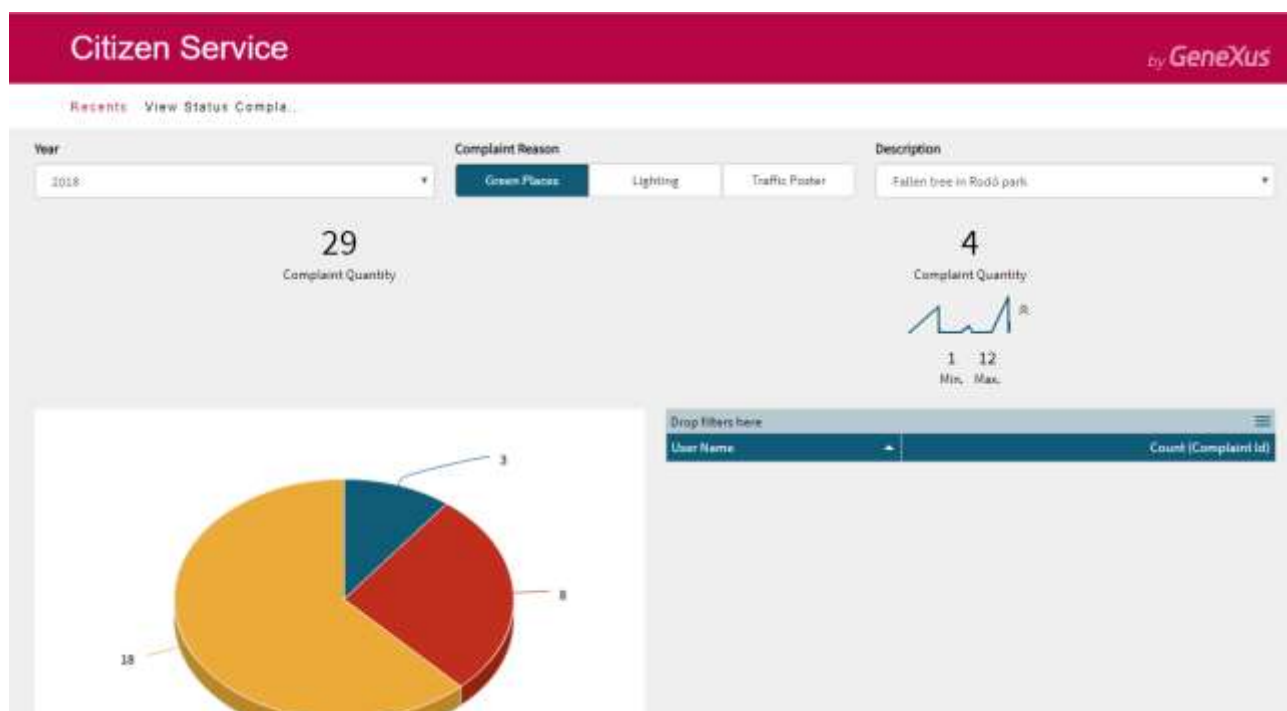
[https://services.odata.org/V4/\(S\(40gwcqlhjmuyfayfnplo0o\)\)/TripPinServiceRW/](https://services.odata.org/V4/(S(40gwcqlhjmuyfayfnplo0o))/TripPinServiceRW/)

## ENTIDADES QUE UTILIZAREMOS

Sob a pasta Entities\_Trns, estão as entidades que modelam a realidade da aplicação. Neste prático, vamos nos concentrar nas reclamações (complaints) dos usuários de acordo com as categorias de reclamações (razões das reclamações, ComplaintReason).

## OBJETO DASHBOARD

Queremos que o backend ofereça uma tela como a abaixo, onde se possam consultar as reclamações (complaints) feitas pelos usuários e filtrá-las por "Year", "Complaint Reason" y "Complaint Description":



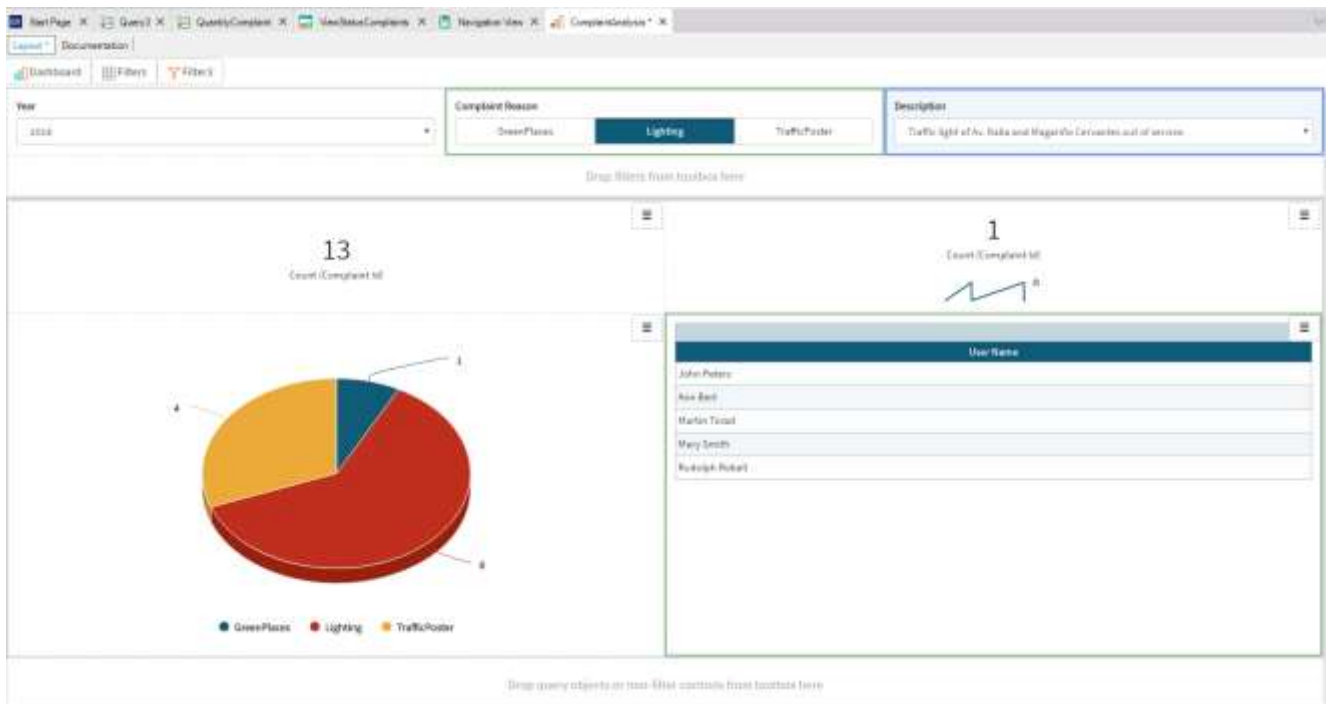
Como você implementa isso? Uma vez feito, teste em execução.

## SOLUÇÃO

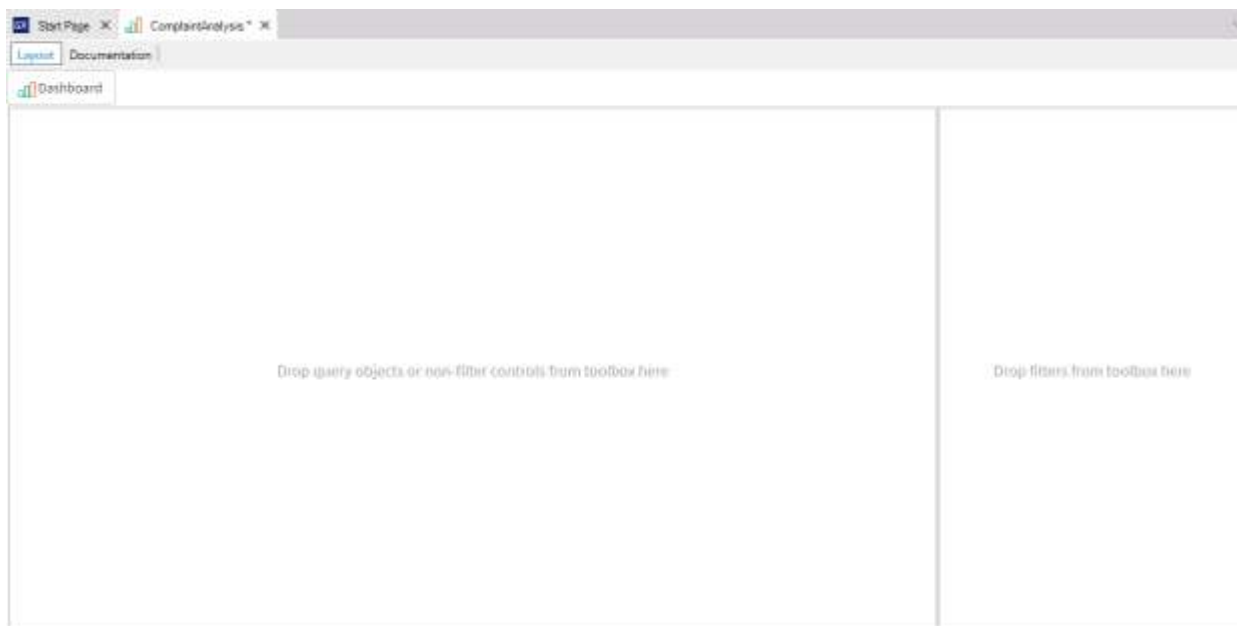
Para isso, vamos criar um objeto do tipo Dashboard (dentro da categoria Reporting) chamado “ComplaintAnalysis”.

Para usar este dashboard, utilizaremos as consultas que já criamos, que estão na pasta “Queries”.

A ideia é ter um dashboard como o mostrado na figura a seguir.



No início, o objeto Dashboard tem duas áreas, a de “Queries” e a de “Filters”



A ideia é colocar as consultas fazendo Drag and Drop na área de consultas, no seguinte local e com o seguinte tipo:

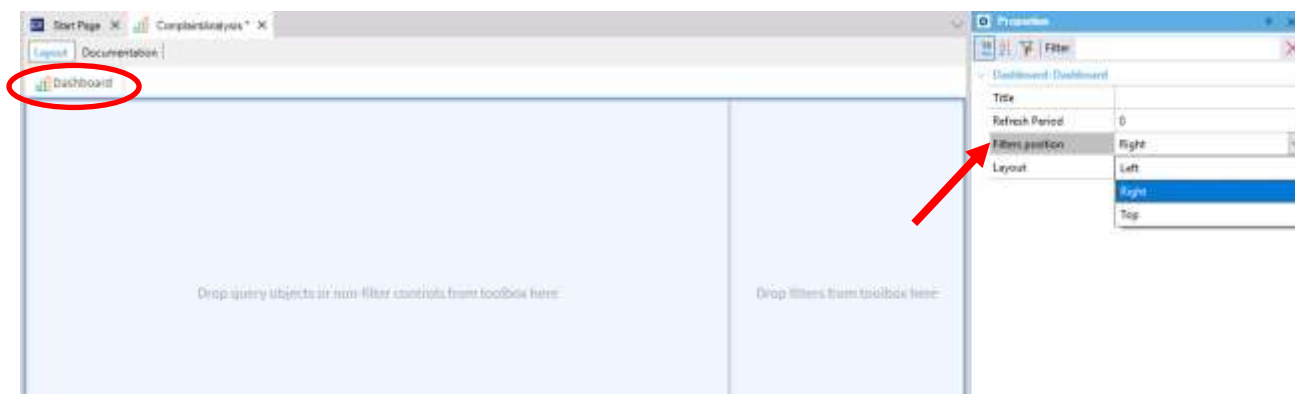
- Ângulo superior esquerdo – QTYComplaintReasonbyYear do tipo Card.
- Ângulo superior direito – QTYComplaint do tipo Card.
- Ângulo inferior esquerdo – QTYComplaintReasonbyYear do tipo Chart.
- Ângulo inferior direito – QTYperUserByYearReasonDesc do tipo Pivot table.

Ao colocar as consultas, GeneXus automaticamente já definiu os filtros que as consultas tinham. Agora você deve mudar o caption dos filtros e modificar o tipo para que eles fiquem como a imagem inicial.

- **“Years”** do Tipo Combo Box estático, com os valores 2017 e 2018
- **“Complaint Reason”** do Tipo Radio Button dinâmico, onde os valores são obtidos do atributo “ComplaintReasonName”
- **“Description”** do Tipo Combo Box dinâmico, onde os valores são obtidos do atributo “ComplaintDescription” e é adicionada a condição “ComplaintReasonName = &ComplaintReasonName”; desta forma ficarão relacionados os Filtros “Complaint Reason” e “Description”

Uma vez finalizada a configuração dos filtros, eles já podem ser modificados, pois há uma live editing do Dashboard.

Dica: Para colocar os filtros na parte superior do objeto Dashboard, você precisa se posicionar na seção Dashboard, lá se habilitam as propriedades e podemos mudar em "Filters position" a posição para Top, em vez de Right.



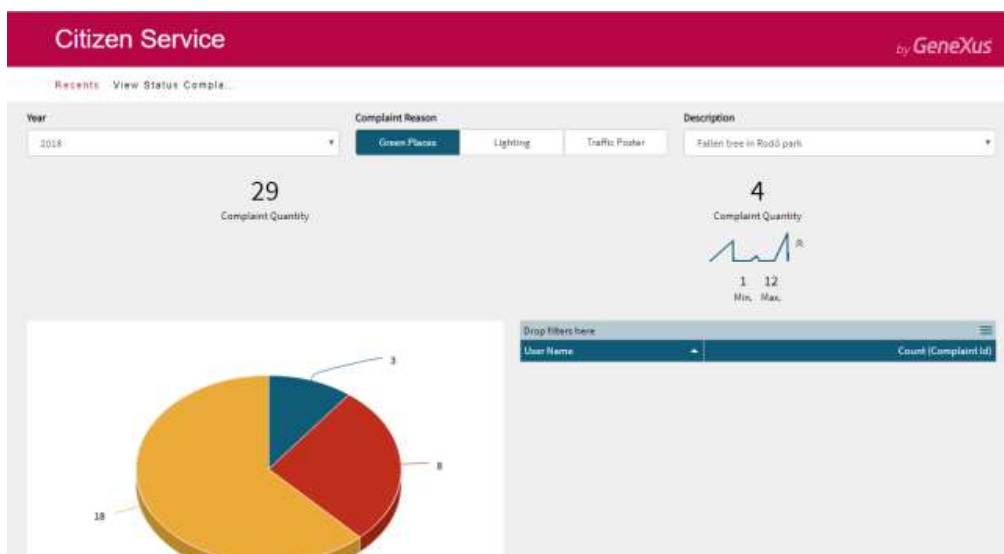
Agora é o momento de adicionar o Dashboard à nossa aplicação. Para isso, vamos criar um Webpanel chamado "ViewStatusComplaints". Nesse webpanel, vamos adicionar um controle do tipo DashboardViewer. Ao controle, vamos associar o objeto dashboard criado anteriormente.

Para finalizar o exercício e poder incorporar o webpanel criado ao backend, podemos adicioná-lo ao objeto "ListPrograms", adicionando as seguintes linhas:

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

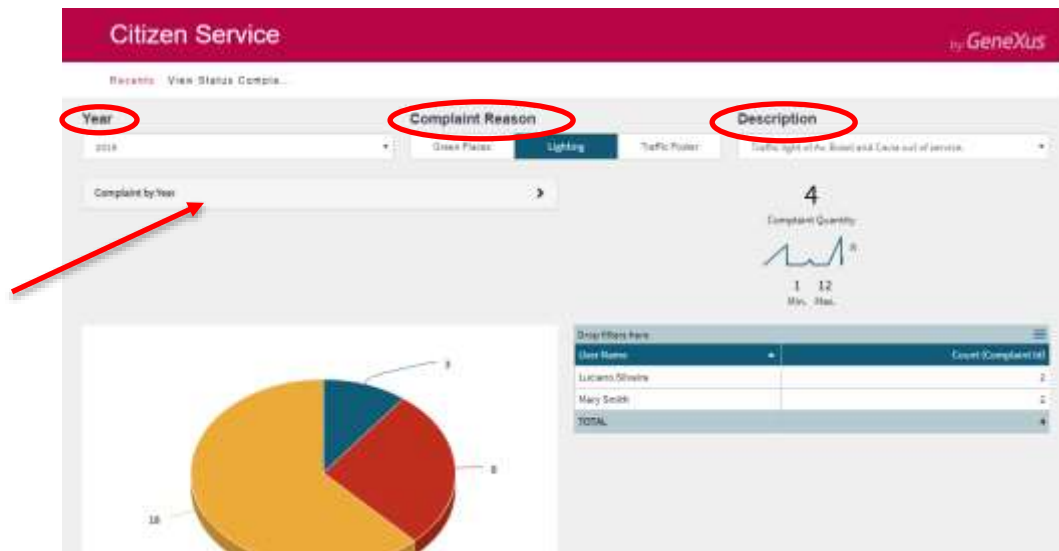
```
Do 'AddProgram'
```

Desta forma, ao executar o Home do nosso backend (Pasta GeneXus/Web), encontraremos a chamada para o nosso "ViewStatusComplaints" e poderemos ver nosso Dashboard em execução:



A ideia agora é modificar os filtros e ver como se comportam as diferentes consultas..

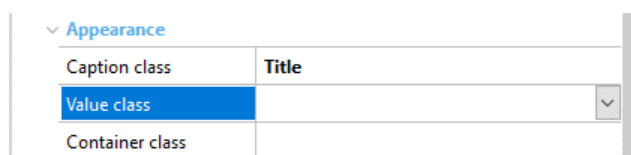
Nesta nova etapa queremos mudar um pouco a estética das consultas, o resultado que queremos alcançar é o mostrado abaixo.



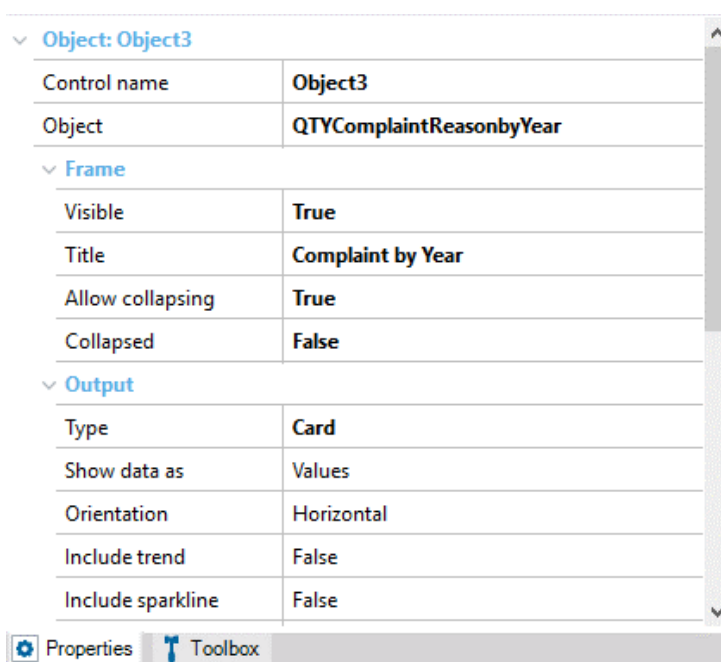
As modificações a fazer são alterar o tamanho do título dos filtros e também que a consulta que está como Card seja mostrada colapsada.



**Tip:** Para alterar a classe associada ao Título, vá para as propriedades em Appearance, Caption class e selecione a classe Title.



Para modificar a consulta `QTYComplaintReasonbyYear`, você precisa alterar o Frame para Visible, colocar um título e habilitar para que possa colapsar.



Depois de todas essas mudanças, salve o dashboard e execute novamente a aplicação, já que estas mudanças não são vistas dentro do GeneXus, mas em tempo de execução..

## KB SOLUÇÃO

Você pode baixar do GeneXus Server a KB solução deste práctico para comparar resultados. É a versão da KB chamada CitizenSolutionPartFour.