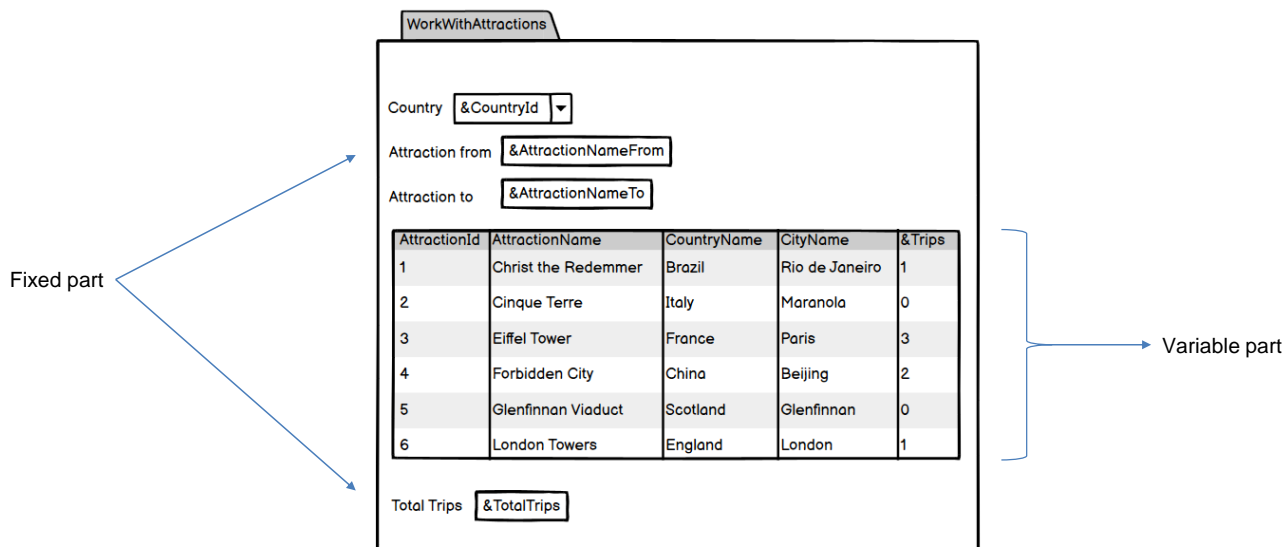


Lógica de carga de dados na tela de um panel

GeneXus™

A seguir veremos como é carregada a tela de um objeto panel, dependendo dos componentes da interface gráfica presentes nela.

Partes de um objeto panel



Em um objeto panel, podemos identificar duas partes diferentes.

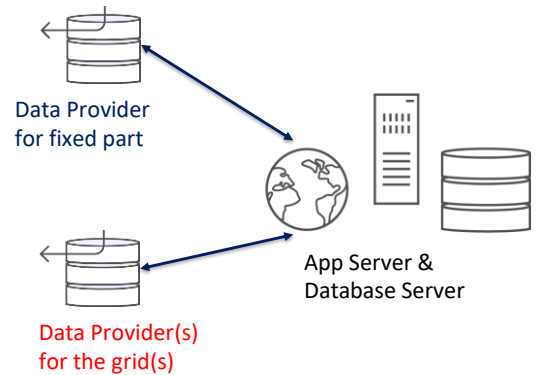
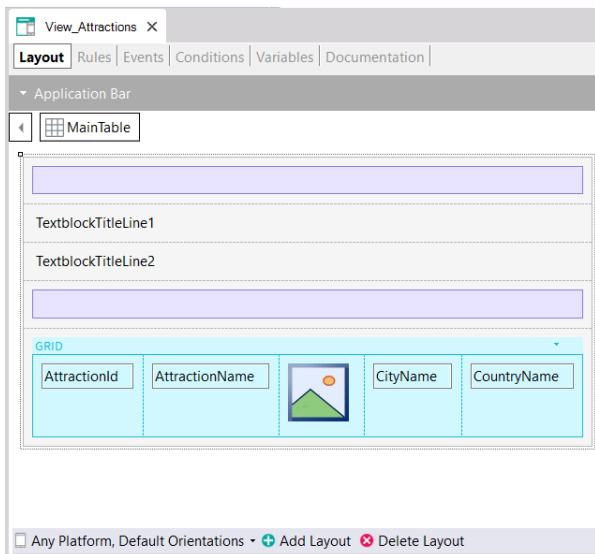
Chamamos de parte fixa ou plana a parte que contém tudo aquilo que está no form e que não está incluído em nenhum grid. Isto inclui controles como textblocks, combos, botões e outros que podemos arrastar a partir da barra de ferramentas.

A segunda parte chamaremos de variável, e é composta pelos grids que foram incluídos no form.

Em um objeto panel sempre teremos uma parte fixa, e podemos ter uma parte variável para cada um dos grids que o panel possua.

Os objetos panel diferem dos webpanels, pois a parte fixa é carregada de forma independente da parte variável.

Partes de um objeto panel

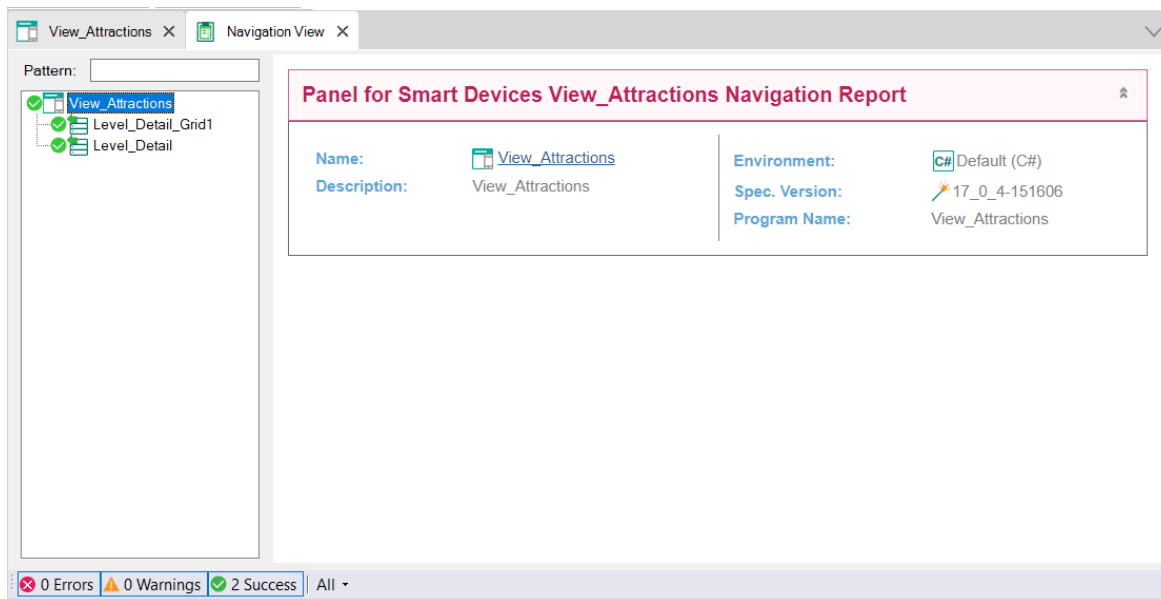


Para cada parte (fixa e grid) GeneXus gerará automaticamente Data Providers independentes que serão publicados como serviços no servidor e serão os que acessarão a base de dados para obter a informação necessária para carregar a parte fixa e a parte variável do panel.

Estes data providers não veremos na Base de Conhecimento, pois GeneXus se encarregará de gerá-los e mantê-los, mas poderemos ver quais dados cada um acessa, se observarmos sua lista de navegação.

Vejamos a lista de navegação do panel View_Attractions que possui em sua parte fixa apenas dois textblocks para as linhas do título e tabelas separadoras e na parte variável um grid com dados de atrações.

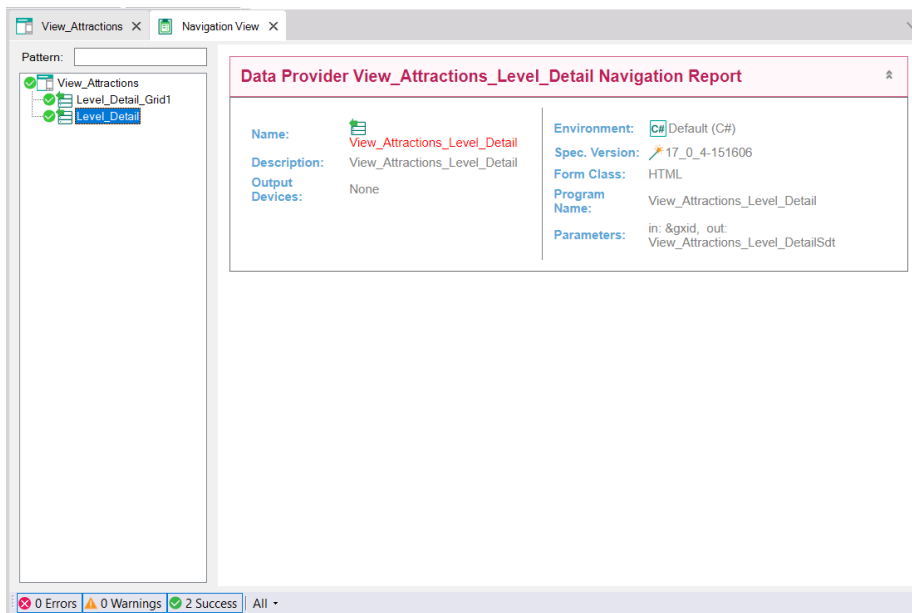
Lista de navegação de um panel



Na lista de navegação do panel `View_Attractions` vemos que sob o nó correspondente ao panel aparecem duas entradas, uma com o nome `Level_Detail_Grid1` correspondente à parte variável do panel que neste caso é composta pelo `Grid1` e outra chamada `Level_Detail` que será a que exibe a carga da parte fixa do panel.

Se o panel tivesse mais de um grid, apareceria um nó de detalhe para cada grid, pois cada um terá uma navegação própria.

Lista de navegação de um panel

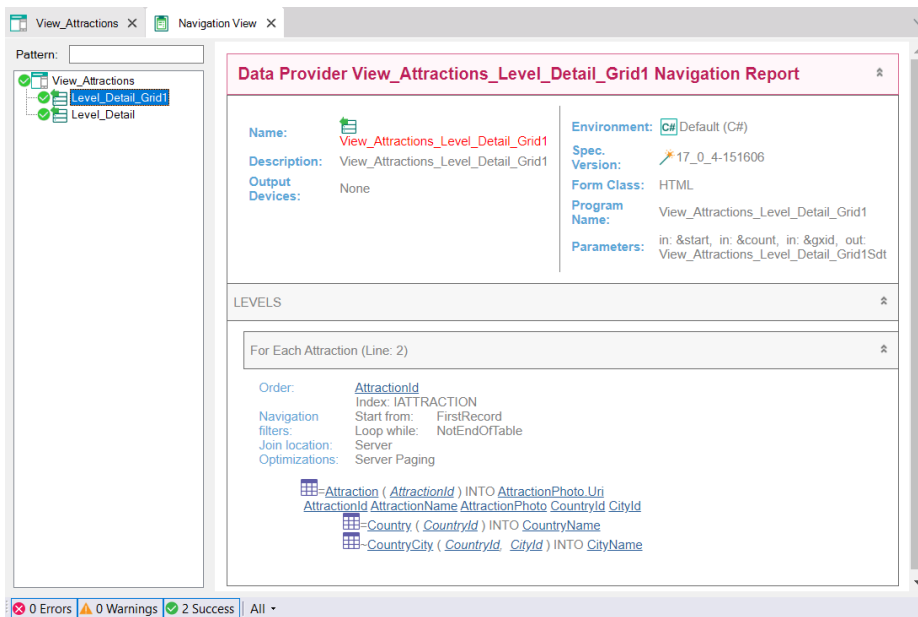


Se selecionamos o nó `Level_Detail`, vemos que é a lista de navegação do data provider `View_Attractions_Level_Detail`.

A lista aparece vazia, pois a única coisa que possui a parte fixa do panel são dois textblocks com dados fixos para o título, não é necessário ir à base de dados para recuperar informações.

Se nesta parte fixa tivéssemos adicionado um Dynamic combo box ou qualquer outro controle que requeira acessar dados da base de dados para exibir informações, aqui apareceria a navegação de acesso a esses dados.

Lista de navegação de um panel



Se agora selecionamos o nó chamado `Level_Detail_Grid1`, vemos que é a lista do Data Provider `View_Attractions_Level_Detail_Grid1`.

Este é o data provider que foi criado automaticamente para carregar o Grid1, publicado como serviço no servidor e invocado pelo panel para acessar a base de dados e recuperar as atrações.

Se repararmos onde diz Environment, observamos que o data provider foi gerado em C# (.NET) como parte do código do backend.

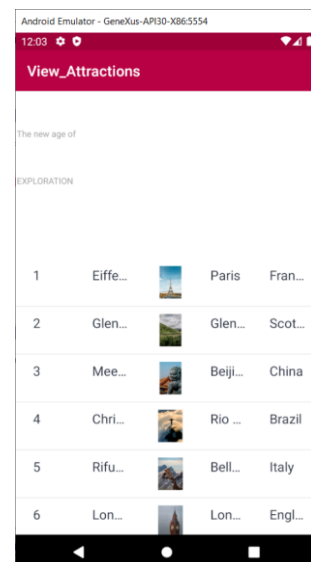
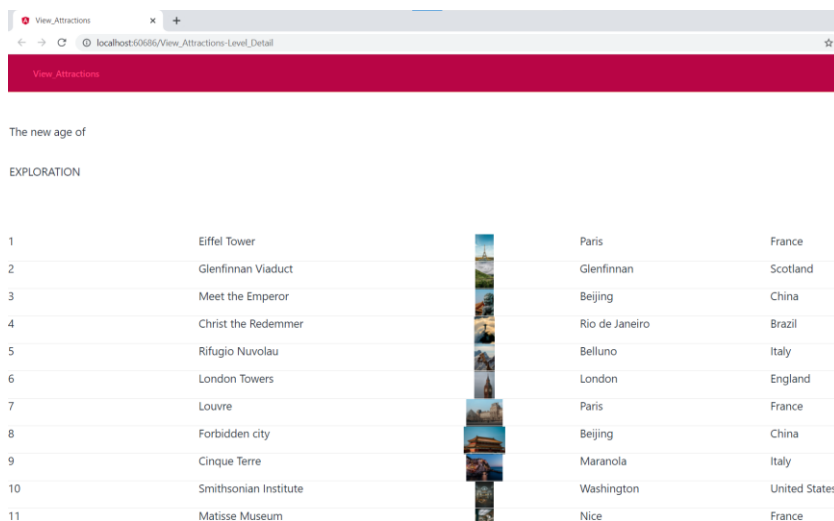
Na seção Parameters, vemos que o data provider recebe alguns dados do grid e devolve carregado um SDT denominado `View_Attractions_Level_Detail_Grid1`, que conterà os dados para carregar o grid.

A informação da lista é semelhante à que veríamos com um web panel com tabela base Attraction, já que o grid do panel está percorrendo a tabela Attraction para mostrar as atrações turísticas e para cada atração acessa a tabela Country para recuperar o nome do país e a tabela City para trazer o nome da cidade da atração.

Uma vez que a parte fixa e as partes variáveis do panel são carregadas separadamente, podemos adiantar que as tabelas base de cada parte são completamente independentes uma da outra. Em outras palavras, em um objeto panel com apenas um grid pode haver 2 tabelas base, uma para a parte fixa e outra para o grid, ao contrário dos objetos webpanel onde, nesse caso, a tabela base será única.

Em outro vídeo, entraremos em detalhes sobre como são determinadas as tabelas base de cada parte de um panel.

Execução do mesmo panel em diferentes plataformas

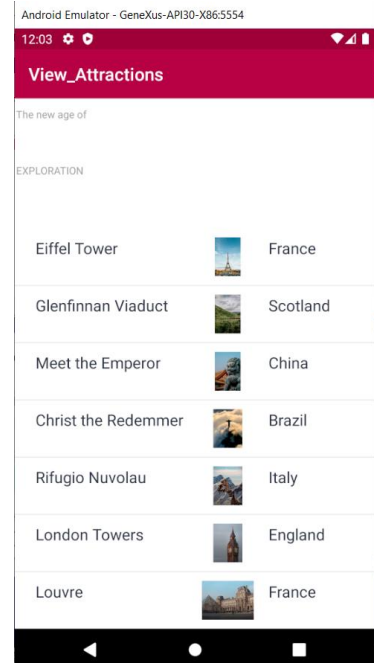
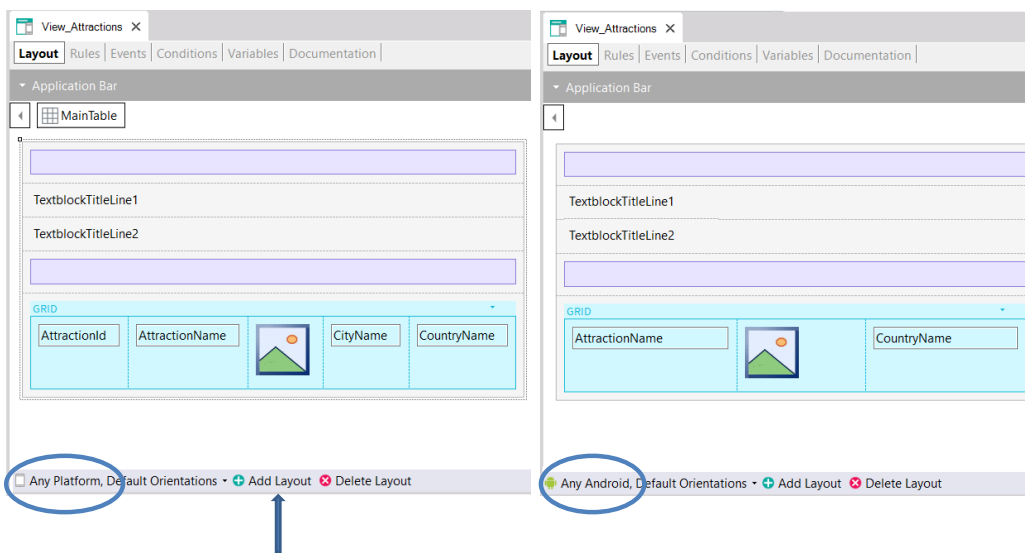


O mesmo objeto panel pode ser executado sem alterações tanto em uma plataforma web, quando gerado por exemplo em Angular, quanto em plataformas nativas móveis quando é gerado em Android ou iOS.

Aqui vemos a execução do mesmo panel em um desktop web gerado com Angular e em um celular Android.

Devido ao espaço disponível, as informações não são vistas iguais em cada caso.

Layouts de um panel



Para que o design de cada tela seja o adequado, temos a possibilidade de criar diferentes layouts para cada plataforma. Cada layout será independente dos outros e em cada um poderemos organizar os elementos da interface gráfica como acharmos melhor.

Também podemos criar vários layouts para uma mesma plataforma, por exemplo, se estou apenas visualizando informações ou se as estou editando e quero que a aparência ou a quantidade de informações exibidas seja diferente em cada caso.

O form de um panel é um container de um ou vários layouts e cada um poderá conter uma certa quantidade e distribuição de controles diferente, mas a lógica de carga de dados na tela será a mesma que a vista anteriormente, ou seja, cada layout poderá ter uma parte fixa e uma ou mais partes variáveis no caso de conter grids.

Portanto, para cada um deles, serão criados de forma independente os data providers necessários para obter as informações da base de dados quando necessário.

Nos próximos vídeos, entraremos em mais detalhes sobre estes temas.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications