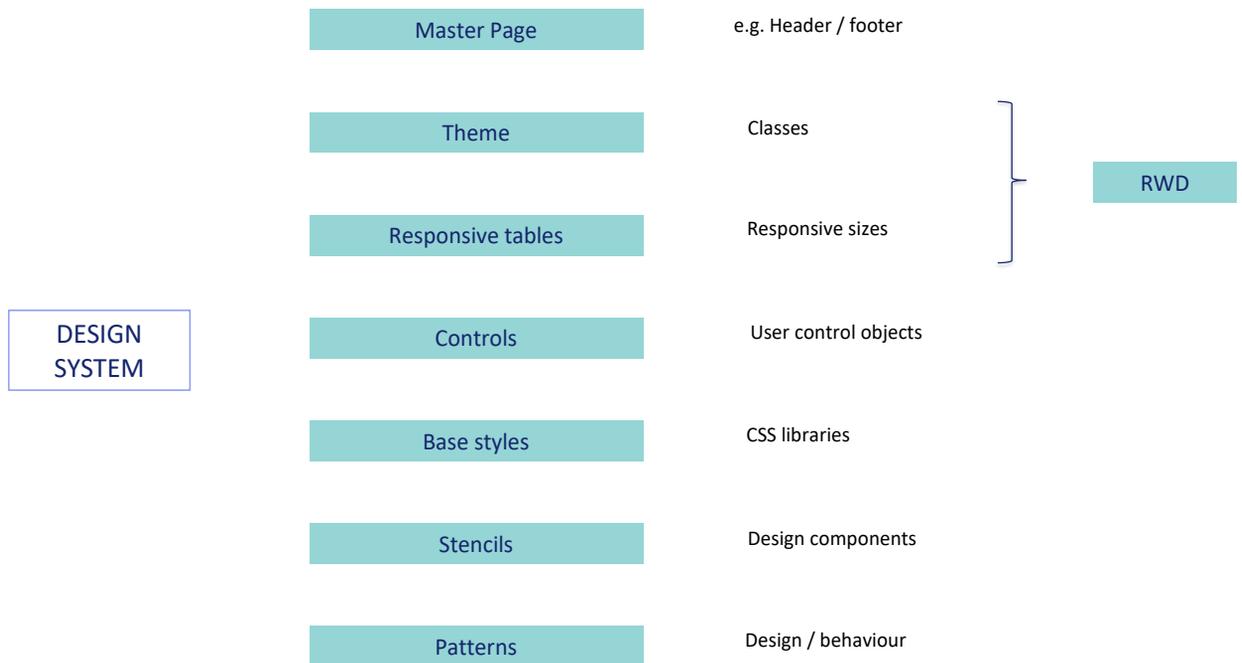


Design System em GeneXus

Default

GeneXus[™]



Neste vídeo, abordaremos superficialmente cada um dos elementos de GeneXus que participam da implementação de um Design System para a aplicação.

Back-office

Attraction Without Parameters

Id: 3

Name: Eiffel Tower

Country Id: 2

Country Name: France

Category Id: 2

Category Name: Monument

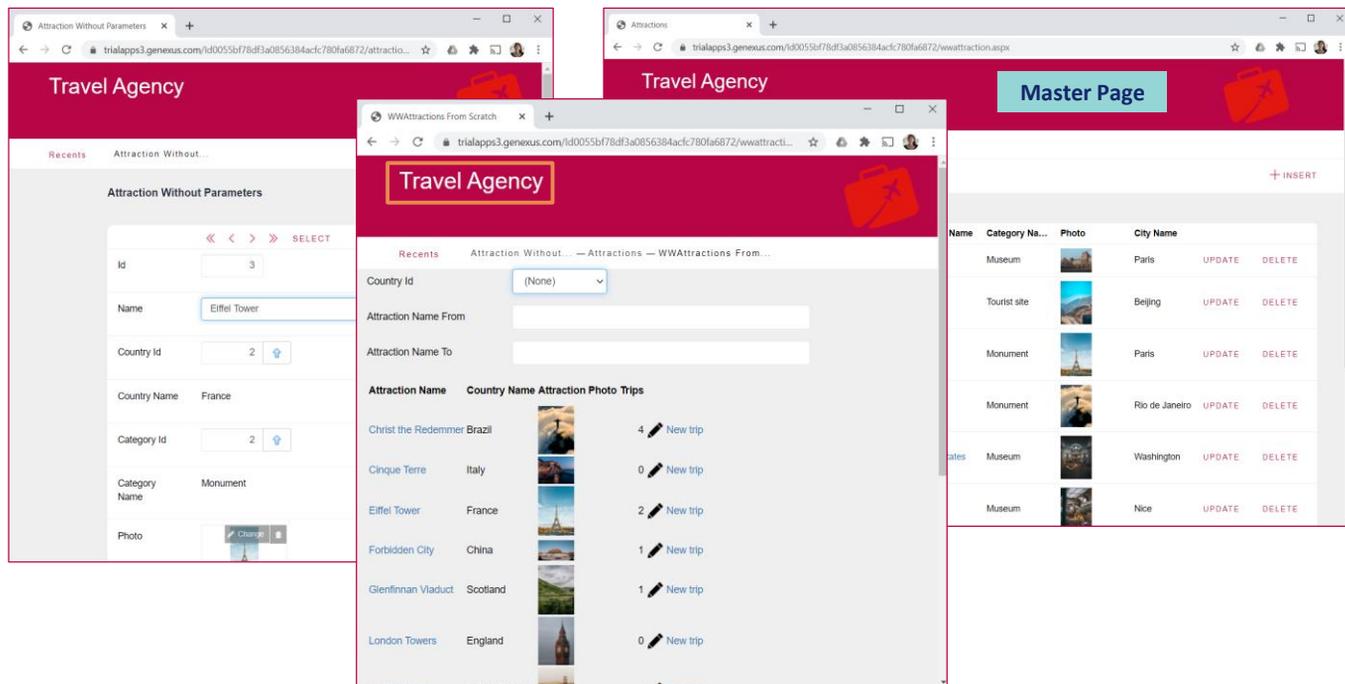
Photo:

Attractions

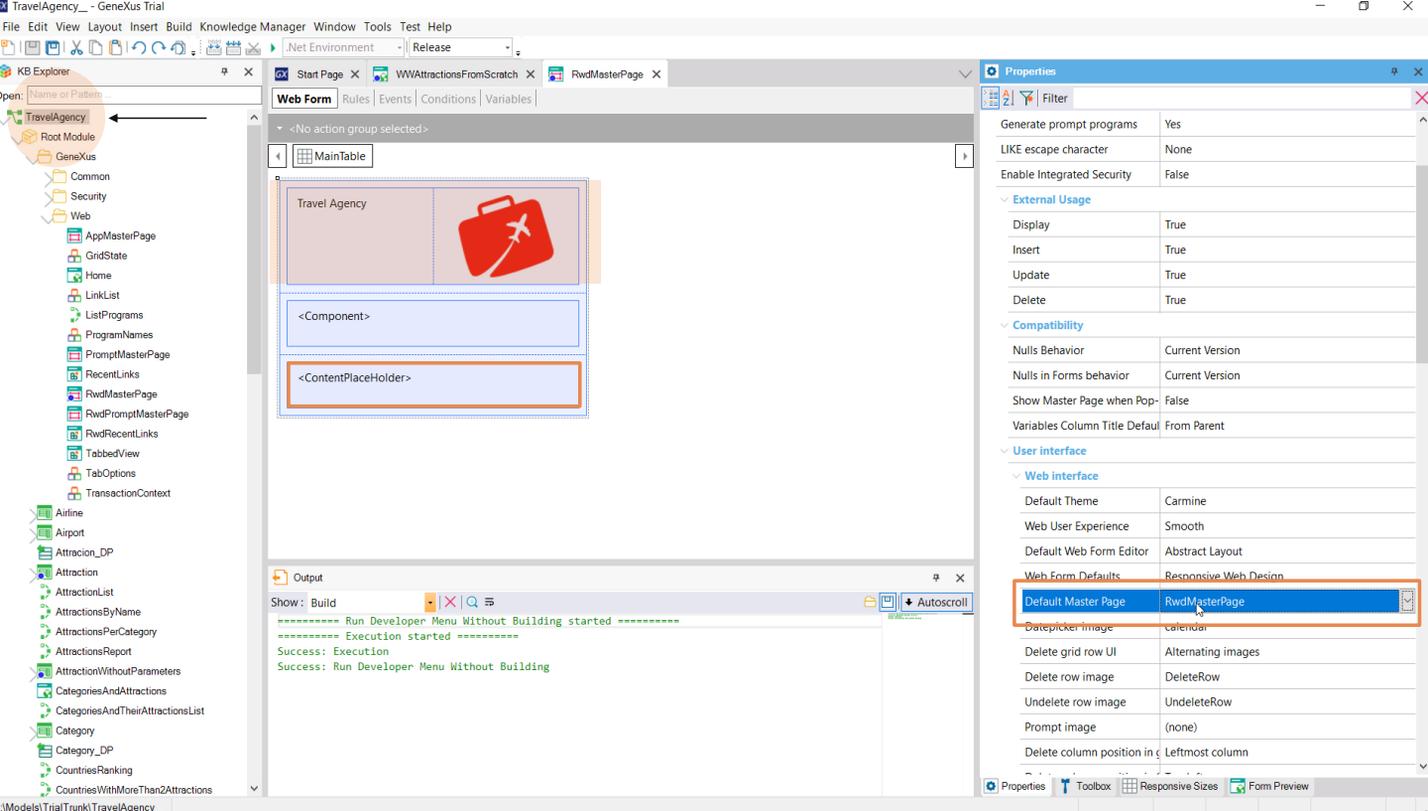
Ordered By: COUNTRY NAME

Id	Name	Country Name	Category Name	Photo	City Name	UPDATE	DELETE
1	Louvre Museum	France	Museum		Paris	UPDATE	DELETE
2	The Great Wall	China	Tourist site		Beijing	UPDATE	DELETE
3	Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
4	Christ the Redeemer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE
5	Smithsonian Institute	United States	Museum		Washington	UPDATE	DELETE
6	Matisse Museum	France	Museum		Nice	UPDATE	DELETE

Se prestarmos atenção às transações que estamos utilizando, e aos painéis criados pelo padrão Work With, vemos que existem elementos que dão uniformidade e coerência, sem nos preocuparmos em conseguir isto. O que significa que GeneXus já está fazendo algo por nós em termos de fornecer um Design System predeterminado.

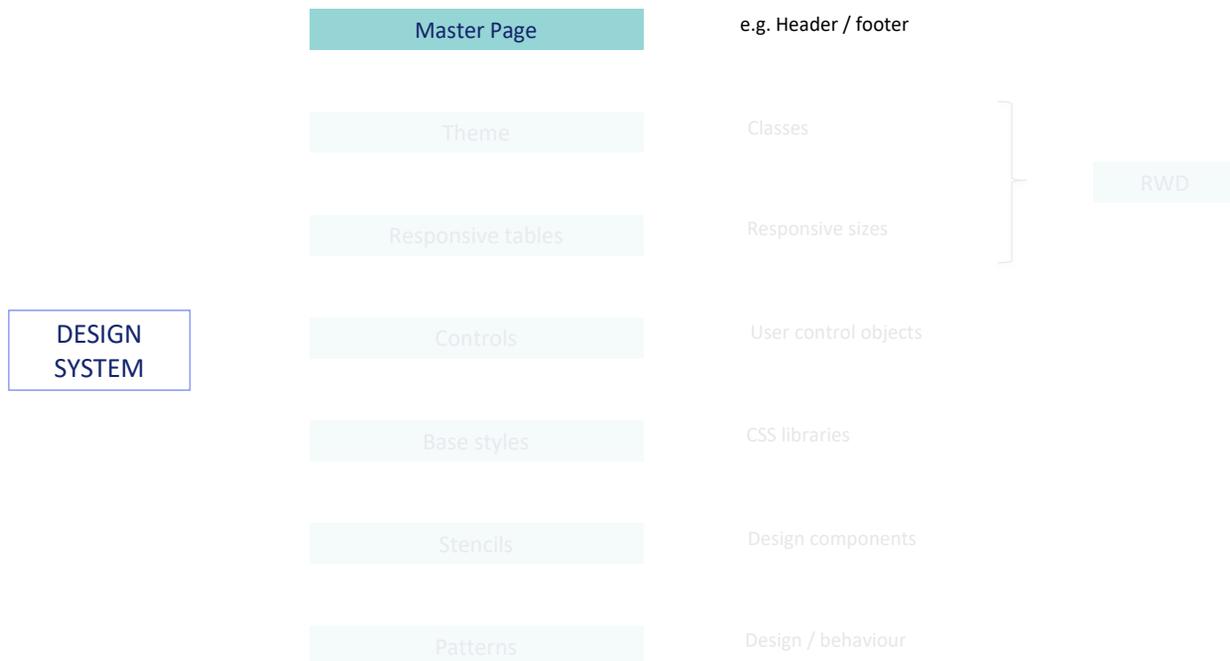


Por exemplo, vemos que todas as telas (inclusive aquelas que criamos do zero, como este web panel) têm um cabeçalho comum. Em nosso caso, nós o personalizamos, mudando o texto padrão para "Travel Agency" e substituindo a imagem que aparecia por esta outra.

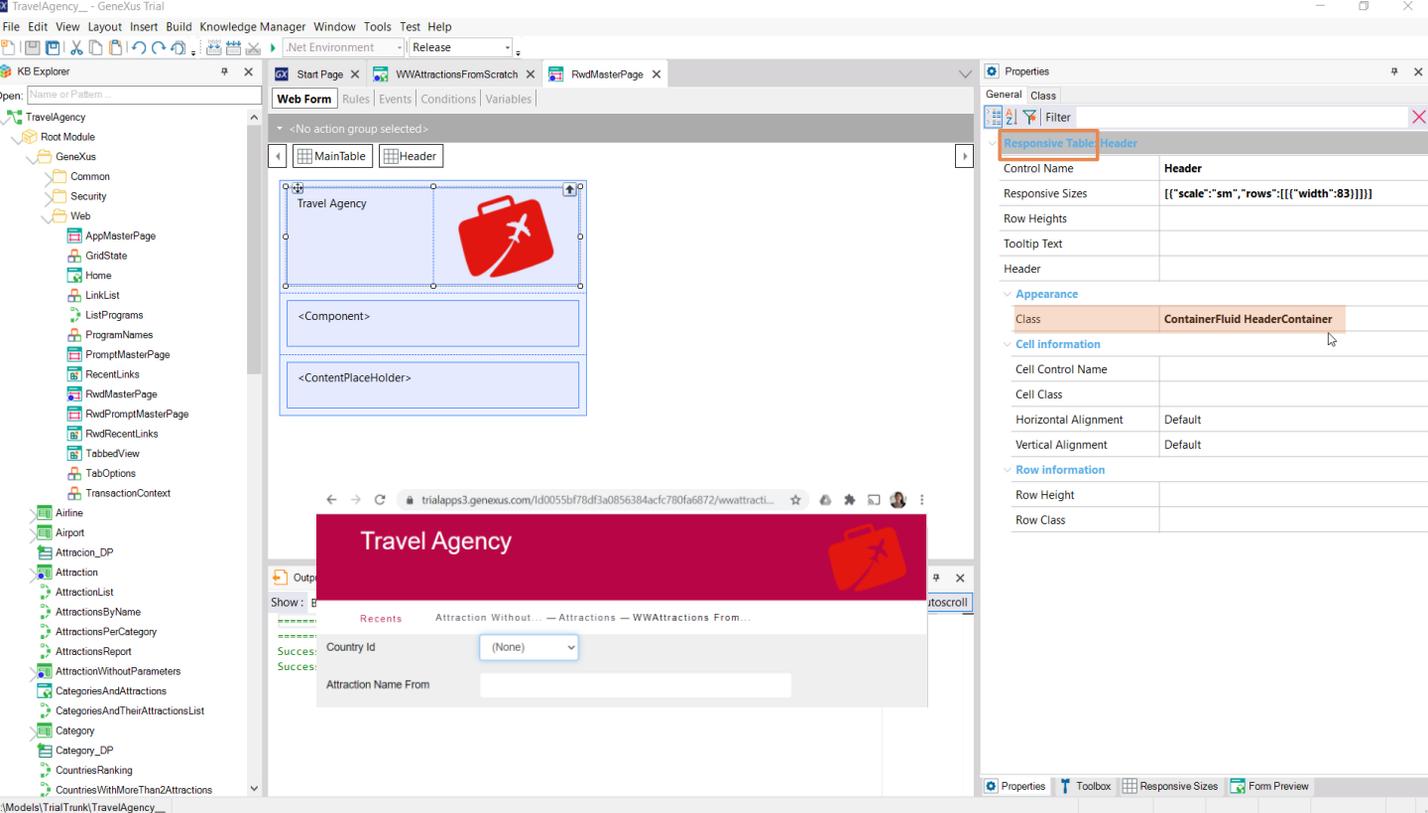


A maneira de obtê-lo é com o objeto web Master Page (Aqui vemos as alterações efetuadas. Lembremos que posteriormente as diferentes telas da aplicação serão carregadas neste controle, o content place holder.).

Por padrão, a versão da KB vem com esta Master Page predefinida, o que significa que ela será aplicada a todos os objetos web criados, como já havíamos visto.

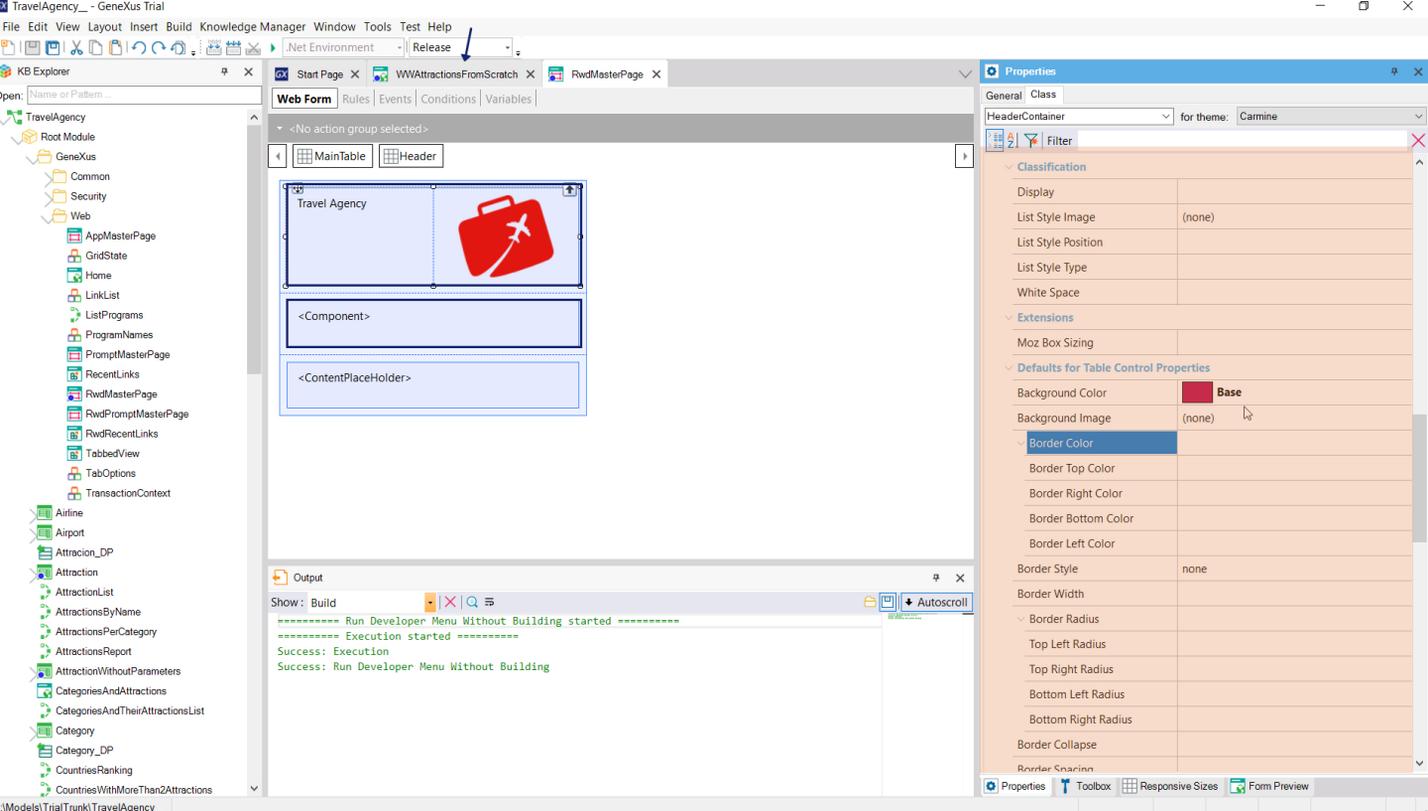


Então, por um lado GeneXus já nos oferece um objeto Master Page para centralizar quais serão as informações comuns que queremos que as páginas compartilhem, **como um cabeçalho ou um rodapé**.



Por outro lado, podemos ver que a cor base, a que se destaca, é um tipo de vermelho, que é utilizado não apenas para o cabeçalho, mas também para os botões e outras ações que são fornecidas ao usuário. Onde se configura tudo isso?

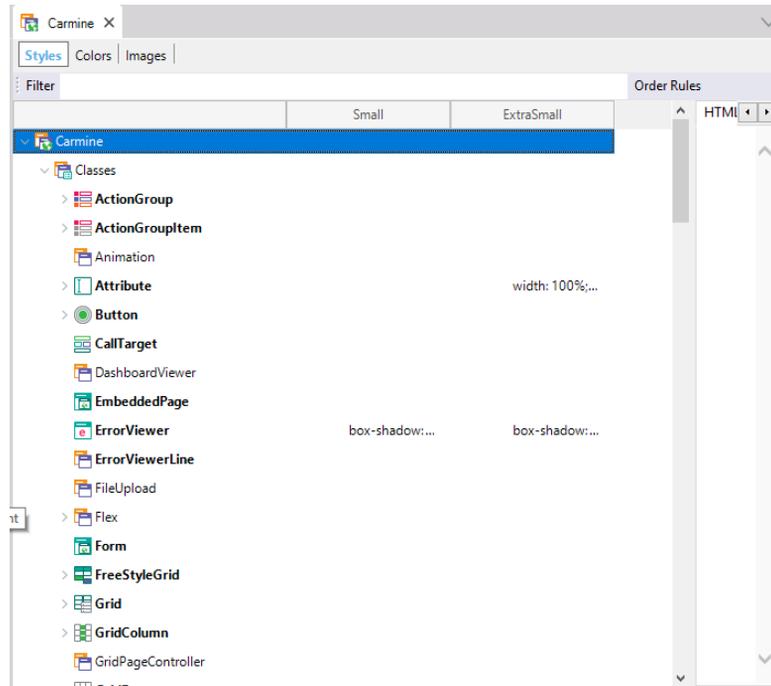
Sabemos que esta tabela tem um fundo vermelho em execução, mas não o estamos vendo! No entanto, observemos que o controle (neste caso, do tipo **tabela responsiva**) tem um par de **classes** definidas, não casualmente sob o grupo “Aparência”.



Se formos para a **guia Class**, vemos que estão sendo editadas as propriedades da segunda das classes associadas a essa tabela. E vemos que **ali está definido, entre outras coisas**, a cor de fundo vermelha que vimos!

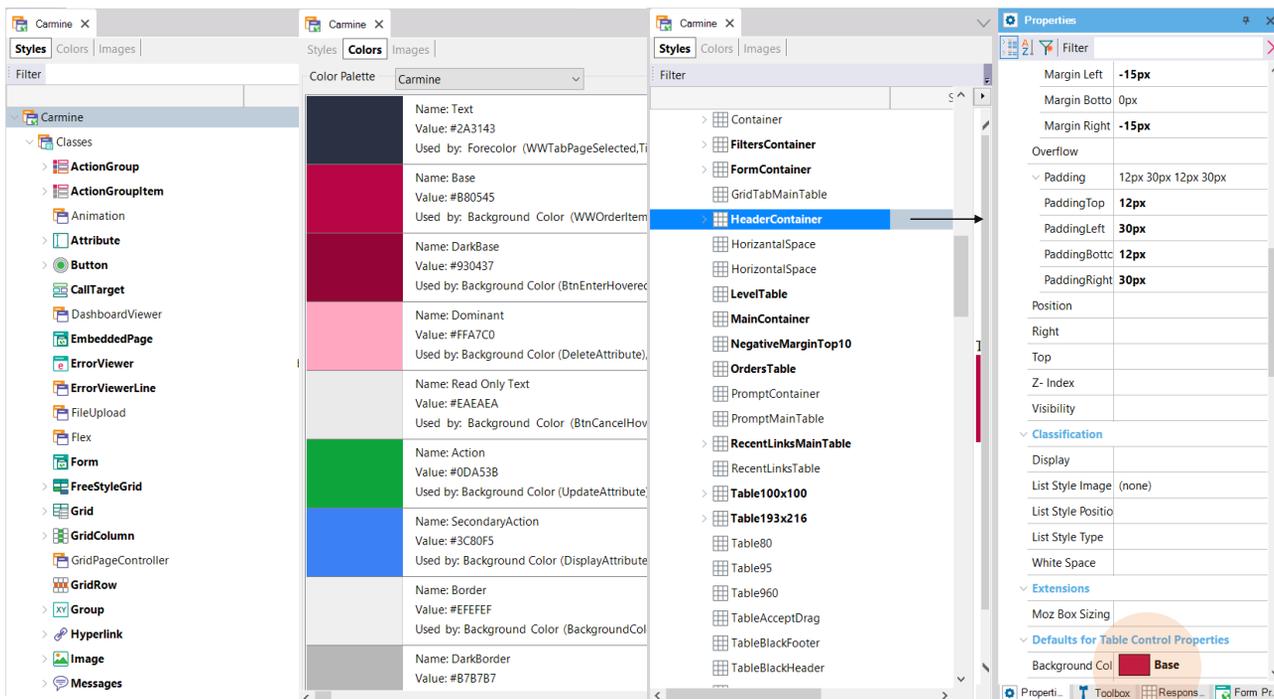
A vantagem das classes é que permitem centralizar o design desse tipo de controle (neste caso, tabela), e que, em nosso caso, outras tabelas que estão no mesmo ou em outros forms possam compartilhar a mesma definição, o que significa que se queremos mudar, por exemplo, a cor de fundo de vermelho para azul para esses controles, não temos que fazer isso um a um em cada controle, ou seja, em cada tabela, mas se faz diretamente na classe e isso já afeta todos os controles que a tenham associada.

Theme



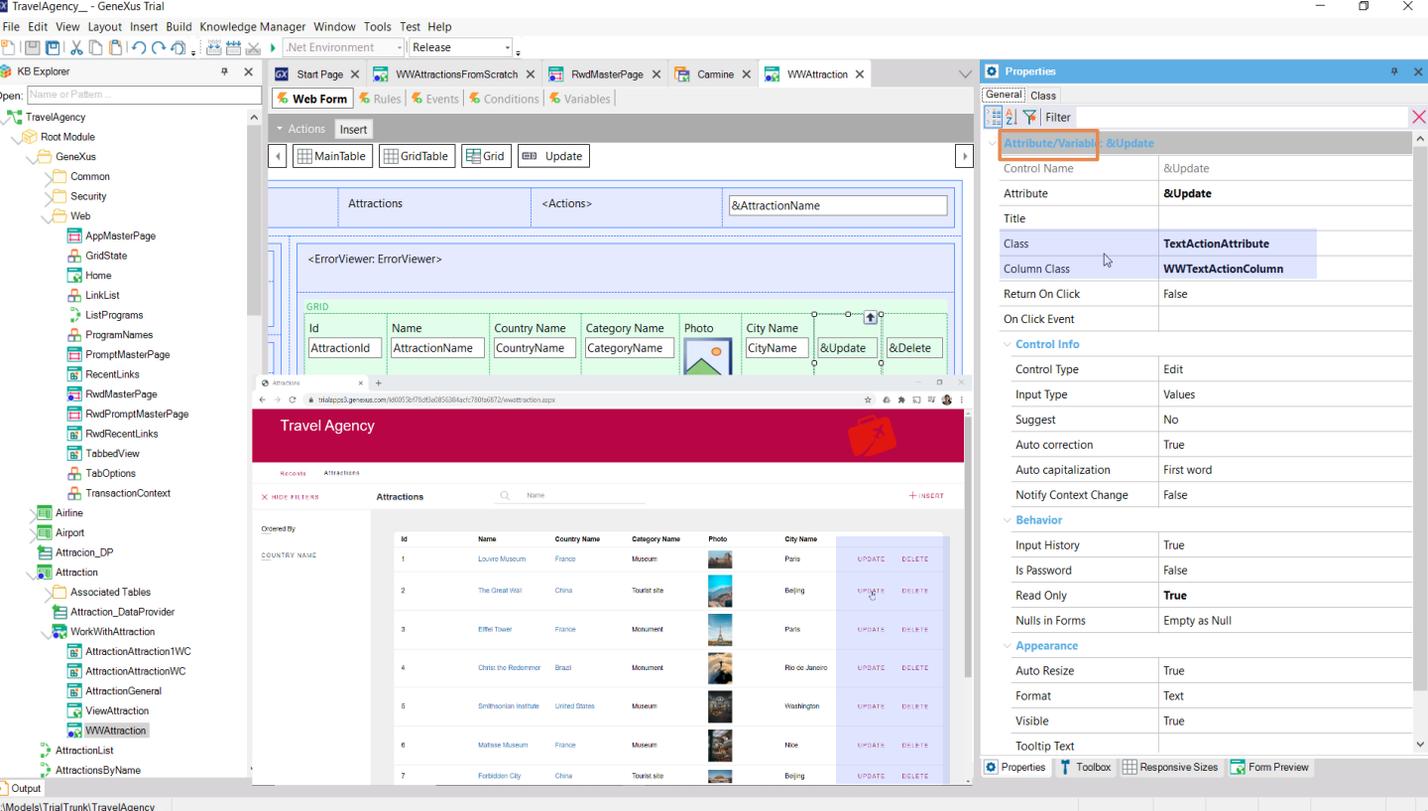
Onde estão todas as classes definidas? No objeto Theme, Tema!

Vemos que, assim como havia uma Master Page predefinida, há também um tema predefinido, chamado Carmine, **que já o vimos repetidamente associado a todos os nossos painéis web.**



Vemos que ao abri-lo, aparecem classes agrupadas por tipo de controle. Aqui é onde está tudo centralizado. E também podemos ver na guia Colors, que mostra a paleta de cores com o mesmo nome... Que é onde está definida a color Base.

Se procuramos por subclasses da classe Table, ali encontramos HeaderContainer, uma das duas classes que tinha associada a tabela da Master Page. O que modificamos em um lado será modificado no outro. Aquilo que vimos era um shortcut para este lugar.



Se formos ver as opções Update ou Delete do pattern Work with, que sabemos terem a color base em execução, vemos que o tipo de controle é Attribute/Variable. E que tem atribuída esta como classe, predefinida pelo Pattern. Podemos aqui mesmo editar suas propriedades sabendo que o que modificamos aqui será modificado no nível central, no objeto Theme, então qualquer outro controle atributo/variável que tenha esta como classe, será modificado.

Em suma, todo controle em um form tem propriedades individuais, que podem ser configuradas apenas para ele... E propriedades que virão da classe que tenha atribuída a ele, que são aquelas que envolvem, mais do que qualquer outra coisa, os aspectos de design, e que serão compartilhadas por muitos controles do mesmo tipo.

Já observamos que neste caso particular em que o controle é, além disso, uma coluna de um grid, aparece outra classe para associar a ele, que é a classe que governa o design e comportamento da coluna.

Attraction Without Parameters

Travel Agency

Attraction Without Parameters

Responsive table

Id: 3

Name: Eiffel Tower

Country Id: 2

Country Name: France

Category Id: 2

Category Name: Monument

Photo:

Attractions

Travel Agency

Grid

Attractions

Ordered By: COUNTRY NAME

Id	Name	Country Name	Category Na...	Photo	City Name	UPDATE	DELETE
1	Louvre Museum	France	Museum		Paris	UPDATE	DELETE
2	The Great Wall	China	Tourist site		Beijing	UPDATE	DELETE
3	Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
4	Christ the Redeemer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE
5	Smithsonian Institute	United States	Museum		Washington	UPDATE	DELETE
6	Matisse Museum	France	Museum		Nice	UPDATE	DELETE

Se observamos novamente o design predefinido da transação e o Work With, vemos que os dados mais relevantes são mostrados em branco sobre cinza, com bordas arredondadas.

Responsive table

The image displays the GeneXus IDE interface for a web form. On the left, a table is shown with the following fields: Attraction Without Parameters, <ErrorViewer: ErrorViewer>, <Toolbar>, Id (AttractionId), Name (AttractionName), Country Id (CountryId), Country Name (CountryName), Category Id (CategoryId), Category Name (CategoryName), Photo, City Id (CityId), and City Name (CityName). The table is highlighted with a blue border, indicating it is a responsive table.

The middle pane shows the Properties window for the selected table, titled "Responsive Table: FormContainer". The properties include:

- Control Name: FormContainer
- Responsive Sizes: [{"scale": "sm", "rows": [{"width": ...}]}]
- Row Heights: [-----]
- Tooltip Text: [-----]
- Header: [-----]
- Appearance:
 - Class: FormContainer
- Cell Information:
 - Cell Control Name: [-----]
 - Cell Class: [-----]
 - Horizontal Alignment: Default
 - Vertical Alignment: Default
- Row information:
 - Row Height: [-----]
 - Row Class: [-----]

The right pane shows the Properties window for the selected table, titled "FormContainer". The properties include:

- List Style Type: [-----]
- White Space: [-----]
- Extensions:
 - Moz Box Sizing: [-----]
- Defaults for Table Control Properties:
 - Background Color: white
 - Background Image: (none)
 - Border Color: [-----]
 - Border Information:
 - Border Top Color: [-----]
 - Border Right Color: [-----]
 - Border Bottom Color: [-----]
 - Border Left Color: [-----]
 - Border Style: [-----]
 - Border Width: [-----]
 - Border Radius: 8px
 - Top Left Radius: 8px
 - Top Right Radius: 8px
 - Bottom Left Radius: 8px
 - Bottom Right Radius: 8px
 - Border Collapse: [-----]
 - Border Spacing: [-----]
 - Lines Font: _ _ 400 14px _ Arial

Isso ocorre porque a tabela onde esses dados se encontram na transação foi atribuída esta classe... que especifica como cor de fundo a branca e um raio para as quatro bordas...

The screenshot shows the GeneXus IDE interface. The main window displays a web form design for 'WWAttraction'. The form includes a search bar for '&AttractionName', a filter section with 'Name' and 'Country' options, and a grid of attraction data. The grid columns are: Id, Name, Country Name, Category Name, Photo, City Name, &Update, and &Delete. The Properties panel on the right shows the 'Grid' control with the following settings:

- Control Name: Grid
- Collection: (empty)
- Base Tm: (empty)
- Order: AttractionName when ...
- Conditions: (empty)
- Unique: (empty)
- Save State: True
- Data Selector: (none)
- Appearance:
 - Class: WorkWith
 - Custom Render: (empty)
 - Empty Grid Text: (empty)
 - Auto Resize: True
 - Width: (empty)
 - Height: (empty)
 - Rows: 10
 - Header: (empty)
 - Tooltip Text: (empty)
- PagingMode:
 - Paging: One page at a time
- Layout:
 - Cell Padding: 1

...e para o grid do Work With esta outra classe, que também especifica branco como cor de fundo, e os mesmos valores que a outra para o raio.

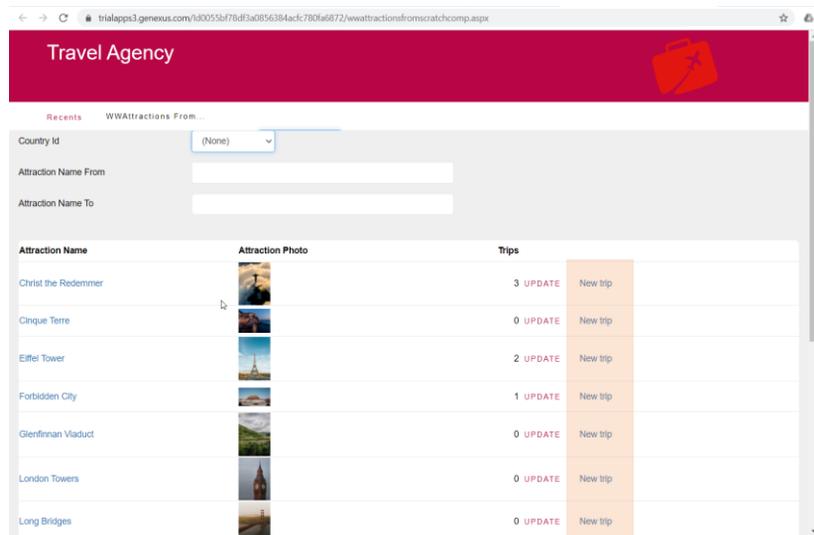
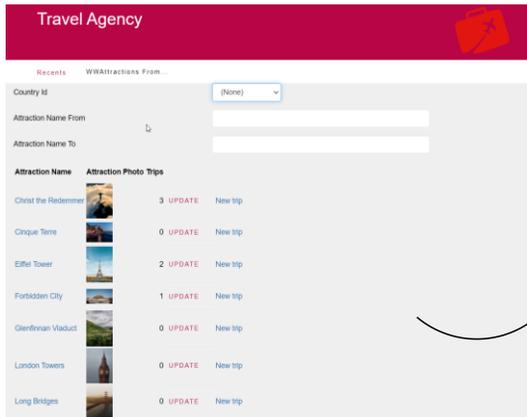
The image shows the GeneXus IDE interface. On the left, the 'Web Form' design view displays a grid with columns: Attraction Id, Attraction Name, Attraction Photo, Trips, &update2, and &newTrip. Below the grid is a 'Total Trips' label with the variable &totalTrips. A green box labeled 'Default Grid' points to the grid. On the right, the 'Properties' window shows the 'Grid: Grid1' class with the following properties:

Property	Value
Control Name	Grid1
Collection	
Base Tm	Attraction
Order	CountryId, AttractionName when not &Coun...
Conditions	CountryId = &CountryId when not &Country...
Unique	
Save State	False
Data Selector	(none)
Appearance	
Class	Grid
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	0

Below the properties window, a preview of the web application is shown. It features a red header with the text 'Travel Agency' and a navigation menu. The main content area displays a list of attractions with columns for Attraction Name, Attraction Photo, and Trips. The list includes items like 'Christ the Redeemer', 'Cinque Terre', 'Eiffel Tower', 'Forbidden City', 'Glenfiddich Vodka', 'London Towers', and 'Long Bridges'. Each item has an 'UPDATE' button and a 'New Trip' link.

Por outro lado, se formos ver o grid do web panel que nós criamos, vemos que tem como classe a de nome Grid, que não tem nada disso configurado... E assim é visto em execução, com todas as colunas coladas, com o fundo default, cinza.

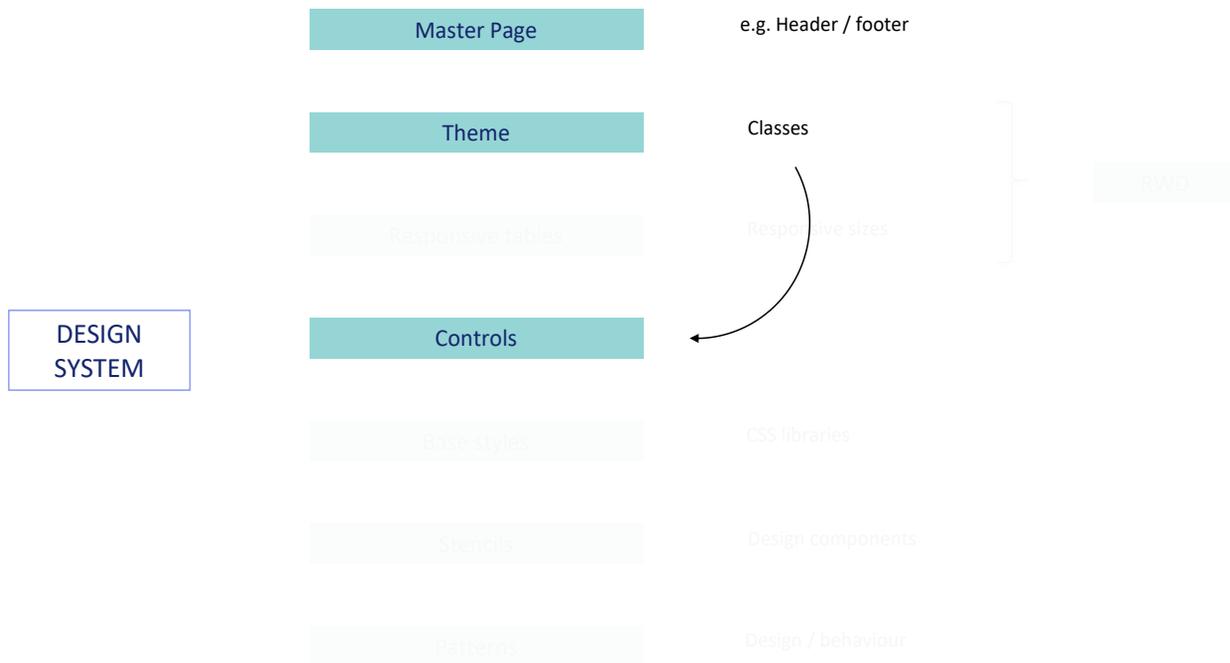
Class: WorkWith



Class	TextActionAttribute
Column Class	
Return On Click	False

Observemos o que acontece se mudarmos sua classe para a mesma que que tem o grid WorkWith...

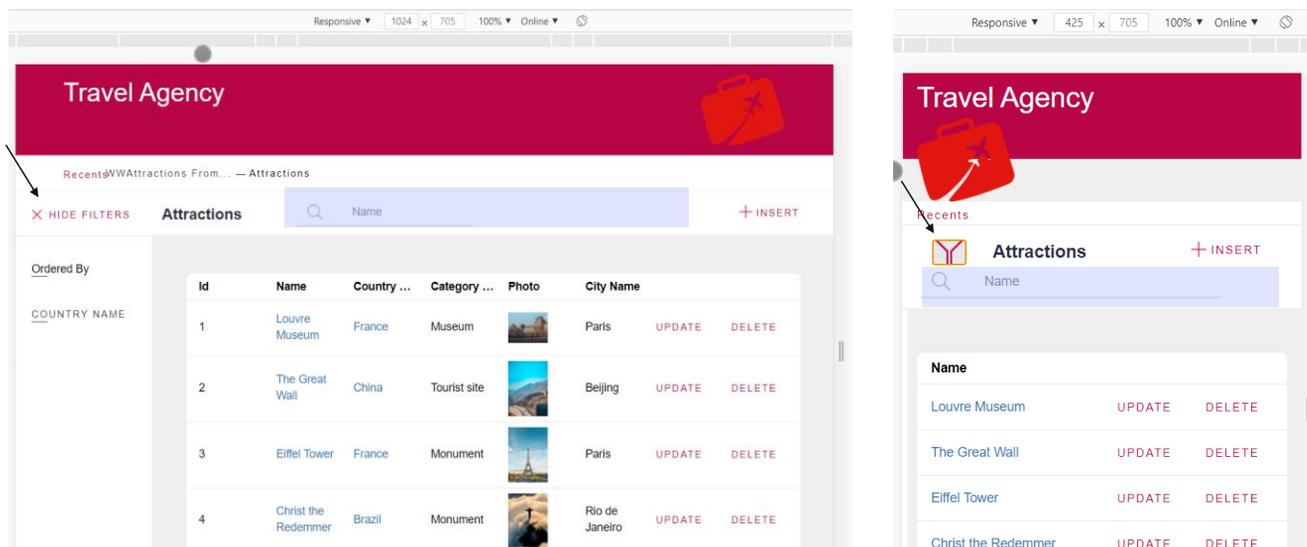
E para manter a uniformidade do Design System, já havíamos mudado para a ação de update, a imagem pelo text block, para que seja idêntico ao do Work With, mas também esta ação do grid deveria se parecer com estas outras...



Podemos deduzir do que vimos que uma porcentagem muito importante do Design System é especificada no objeto Theme, através de suas classes (as predefinidas e as que adicionamos). Estas classes são aquelas que são atribuídas aos controles do form, permitindo assim abstrair o design, estabelecer uma lógica e desta maneira padronizá-lo de acordo com a função que assume cada controle no todo.

Portanto, uma parte fundamental do design será identificar quais classes serão necessárias para os controles e aplicá-las.

Responsive Web Design (RWD)



Por outro lado, voltando ao predefinido (lembramos que alguns navegadores nos oferecem um modo de visualização –neste caso pressionando F12– que nos permite variar o tamanho da tela para ver como ficaria [naquele tamanho](#))...

Podemos ver que, se reduzirmos a tela do Work with de atrações o suficiente para corresponder à tela de um telefone, os filtros serão exibidos de forma diferente. E, por exemplo, o campo para o qual o usuário possa inserir um filtro por nome de atração aparece abaixo do título e não à sua direita. Além disso, os recent links aparecem como menu suspenso, em vez de expandidos. E a imagem da master page está aparecendo abaixo e não ao lado.

E com a diminuição da largura da tela, quase todas as colunas também desaparecem, com exceção da que mostra o nome e as ações de Update e Delete..

Este comportamento é conhecido como “responsivo” e ao design web que o considera como Responsive Web Design. Hoje em dia, não é aceitável um design que não adapte o que se vê ao tamanho de tela.

Uma parte da responsividade é estabelecida através das **tabelas responsivas**...

Responsive tables

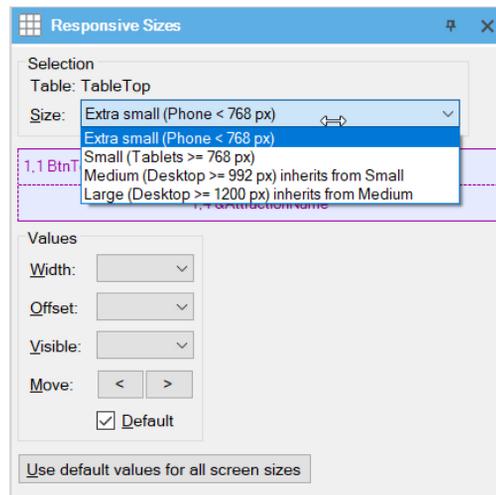
The screenshot displays the GeneXus IDE interface. The main workspace shows a design for a responsive table. At the top, there are filter controls: 'Hide Filters', 'Attractions', '<Actions>', and '&AttractionName'. Below this is a header section with a red background and a white icon of a suitcase, containing the text 'Travel Agency'. Underneath the header is a table with columns for 'Name' and 'Country'. A search bar is visible with the text '&CountryName'. Below the search bar is a 'Recents' section with a search icon and the text 'Name'. At the bottom, there is a table with a row containing 'Louvre Museum', 'UPDATE', and 'DELETE' buttons.

The Properties panel on the right shows the configuration for the 'TableTop' control. The 'Responsive Sizes' property is set to `{{"scale":"xs","rows":{{"width":...`. The 'Appearance' section shows the 'Class' property set to 'TableTopSearch'. The 'Cell information' section shows the 'Cell Class' property set to 'TableTopSearch'. The 'Row information' section shows the 'Row Height' and 'Row Class' properties.

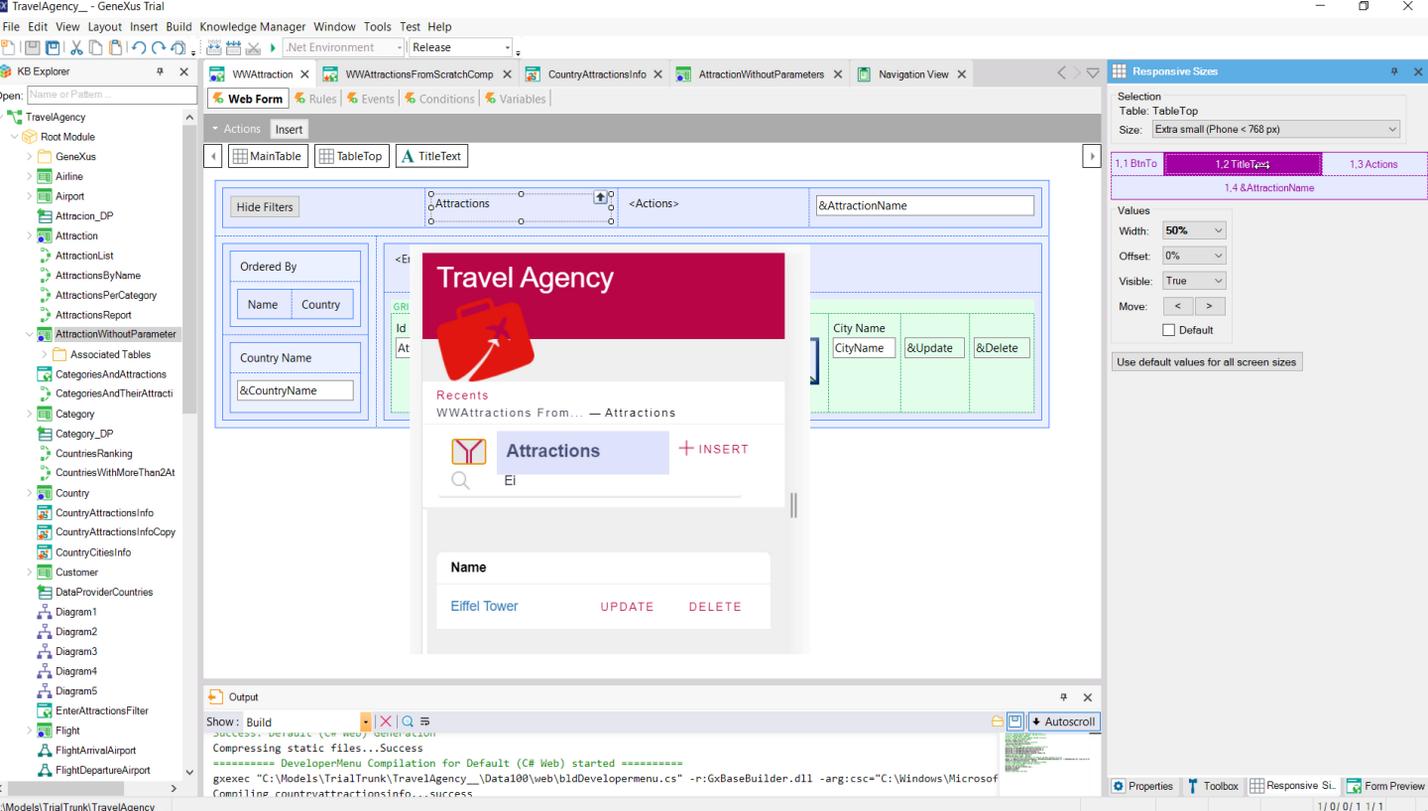
Por exemplo, esta tabela **do Work With de atrações que criou o pattern**, que vemos que é responsiva, contém estes quatro controles. **O primeiro corresponde aos filtros adicionais**, o segundo corresponde a este textblock, o terceiro à **ação de Insert**, e o último corresponde à variável de filtro que vimos.

Se editamos as propriedades da tabela veremos esta, “Responsive Sizes” (tamanhos responsivos).

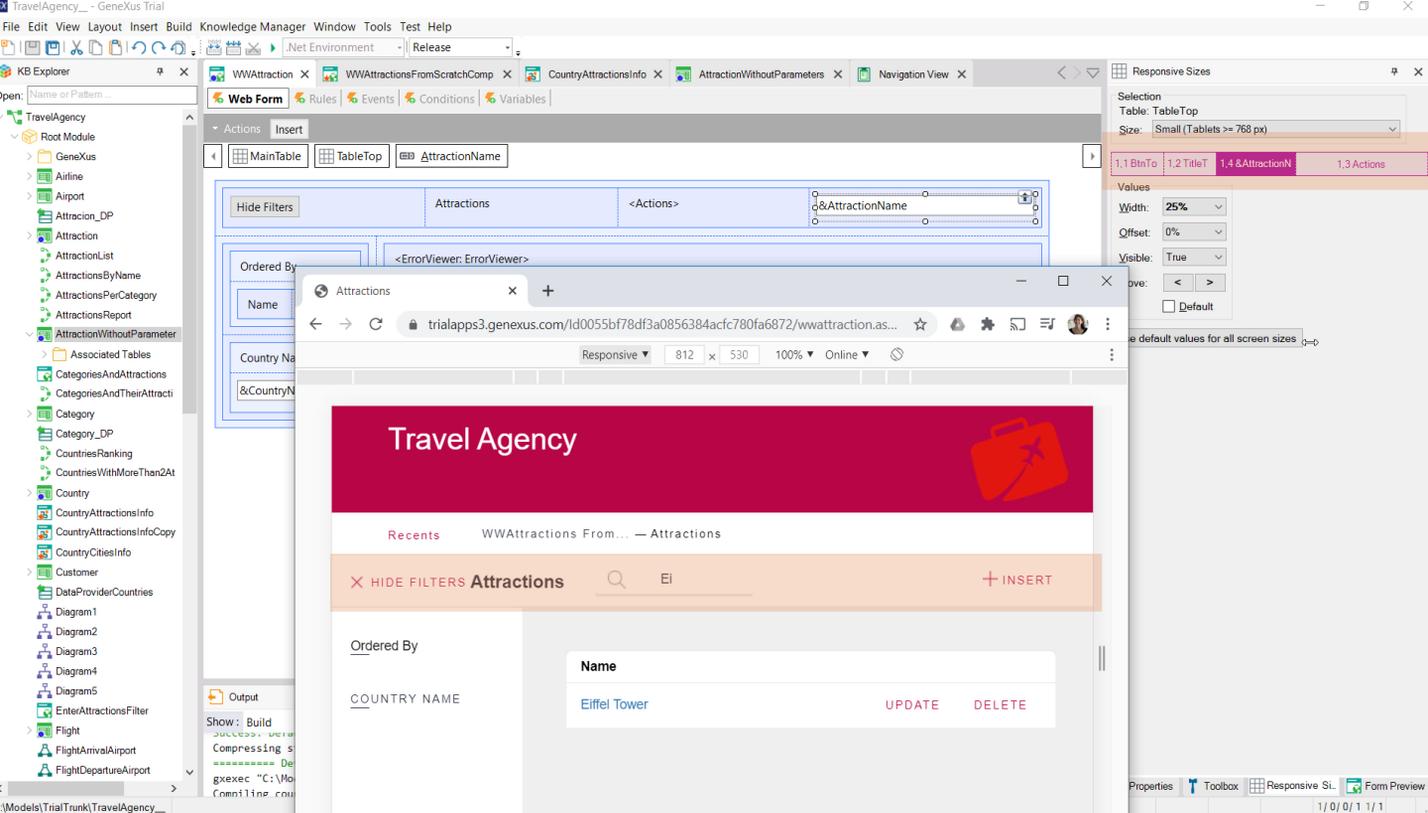
Responsive tables



Aqui se estabelece como se deseja visualizar o conteúdo da tabela de acordo com quatro tamanhos de tela: **Extra small**, que corresponde a telefones; **Small**, que corresponde a tablets; **Medium** que corresponde a monitores de notebooks ou Pcs menores que 1200 pixels; e **Large** que corresponde a tamanhos maiores.



Aqui podemos ver que se o tamanho for **Extra small**, aparecerá primeiro o botão para habilitar os demais filtros, ocupando 17% da largura da tela, a seguir à sua direita virá o título, que ocupará 50% da largura restante, e por último as ações, que aqui será apenas o Insert, ocupando os 33% restantes, e a variável de filtro aparecerá abaixo de tudo isso, ocupando 100%.

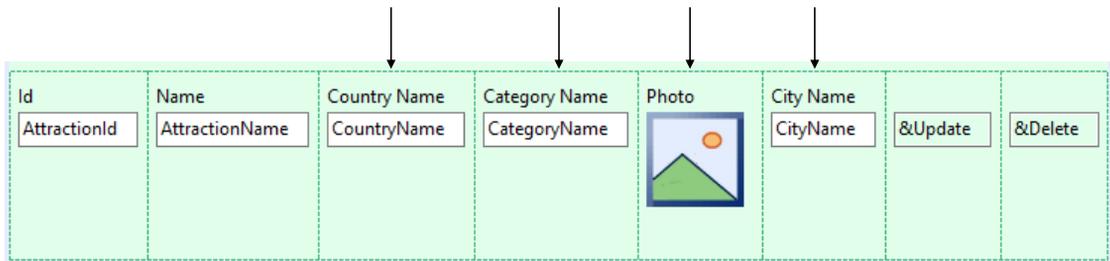


Enquanto que se o tamanho for **Small**, estará tudo em uma mesma linha, com botão, título, variável e ação, nessa ordem.

Responsive Web Design (RWD)

- Responsive Sizes of Responsive tables
- Rules of Theme: Small, Extra-small and Default

ie: to hide grid columns if Extra small or Small size



Id	Name	Country Name	Category Name	Photo	City Name	&Update	&Delete
AttractionId	AttractionName	CountryName	CategoryName		CityName		

Dissemos que uma parte importante da responsividade, a maior, é obtida fazendo uso destas tabelas e estas propriedades. Mas outra parte, **mais sutil**, é alcançada através das **classes**.

The screenshot shows a web form editor with a grid control. The grid has the following columns: Id, Name, Country Name, Category Name, Photo, City Name, and actions (&Update, &Delete). The Properties window on the right shows the configuration for the 'CountryName' attribute, with the class set to 'WWColumn WWOptionalColumn'. Arrows point from the 'Country Name' column in the grid to the 'CountryName' property in the Properties window.

Attribute	Country Name
Title	Country Name
Class	Attribute
Column Class	WWColumn WWOptionalColumn
Return On Click	False

Control Info	
Control Type	Edit
Input Type	Values
Suggest	No
Auto correction	True
Auto capitalization	First word
Notify Context Cha	False

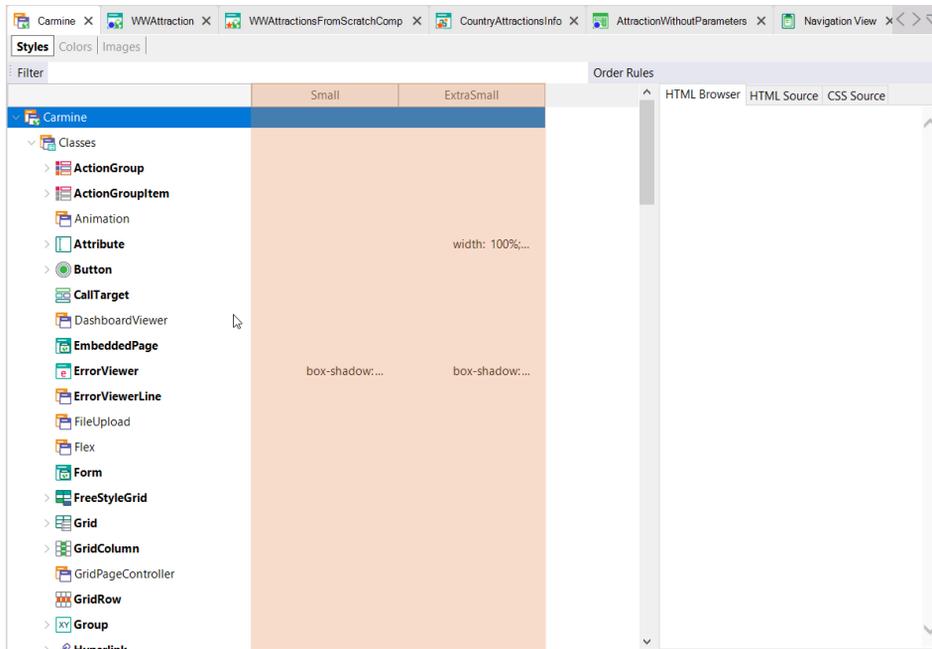
Behavior	
Input History	True
Is Password	False
Read Only	False
Nulls in Forms	Empty as Null

Appearance	
Auto Resize	True
Format	Text
Visible	True
Tooltip Text	
Invite Message	

Por exemplo, ocultar colunas do grid dependendo do tamanho da tela.

Vemos que a classe da coluna do nome da atração é WWColumn, mas que para as outras colunas é adicionada, além disso, a classe WWOptionalColumn.

RWD: Theme rule columns

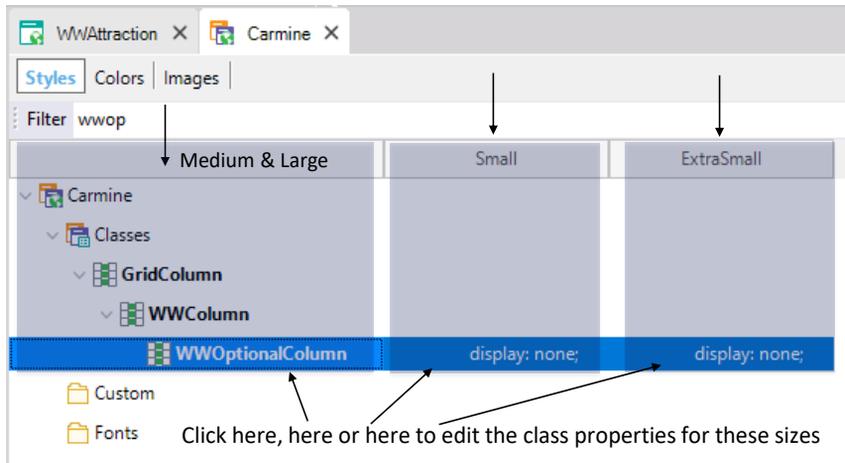


Se formos ao theme, vemos que estão aparecendo estas duas colunas, que o que fazem é nos permitir variar os valores das propriedades da classe, de acordo com o tamanho da tela.

Por padrão, apenas dois são definidos: Small e ExtraSmall. Mas na verdade temos três, já que os valores que vemos na coluna default correspondem a Medium e Large.

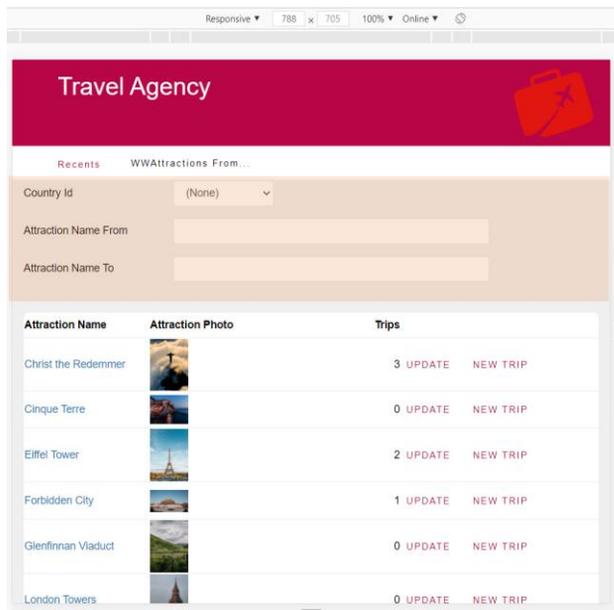
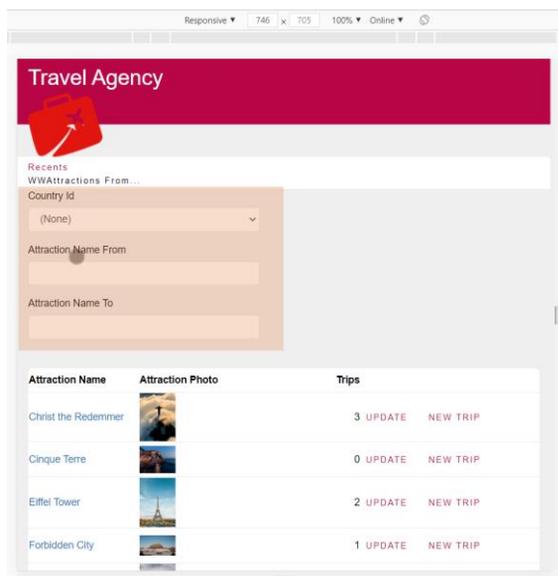
RWD: Theme rule columns

- Rules of Theme: Small, Extra-small and Default



Então, aqui podemos ver que, enquanto a propriedade Display para esta classe não diz nada, o que significa que será exibida em execução para tamanhos de tela Medium e Large, para tamanho Small e ExtraSmall, tem o valor "none". E é por isso que não são exibidas.

How to improve responsiveness in our web panel



Por outro lado, no web panel que implementamos do zero, não temos este comportamento responsivo para o grid. Ao diminuir o tamanho da tela, continuamos vendo as mesmas colunas.

A única responsividade predefinida que temos é a que vem da Master Page, onde para tamanho Extra-small a imagem está ficando abaixo e não à direita e a lista de links recentes aparece como menu vertical, e para o nosso web panel propriamente dito, a que que envolve os campos de filtro, que estão ocupando 100% da largura com sua etiqueta acima, quando para tamanhos Small em diante, ficam à esquerda.

How to hide columns when Small or Extra-small

Travel Agency

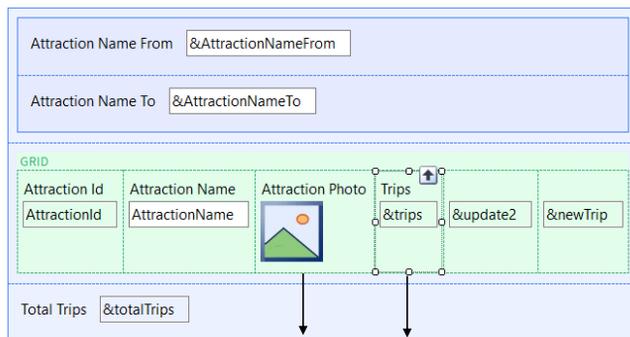
Recents WWAttractions From...

Country Id (None) ▾

Attraction Name From

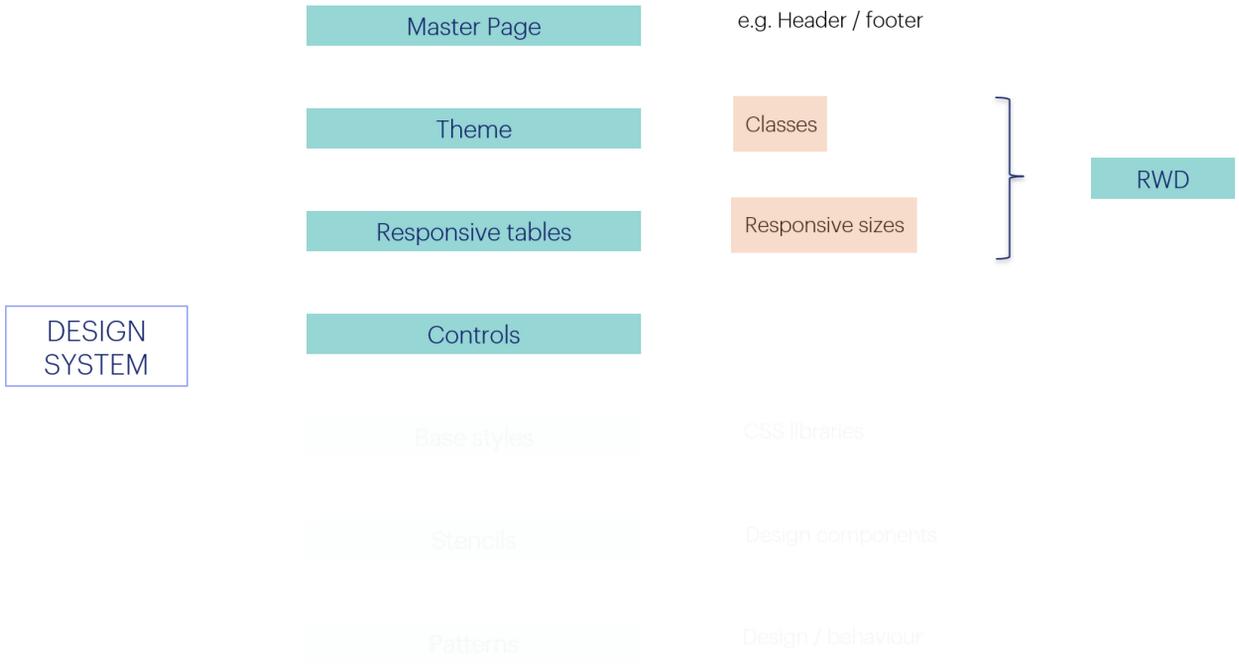
Attraction Name To

Attraction Name	Attraction Photo	Trips	
Christ the Redemmer		3	UPDATE NEW TRIP
Cinque Terre		0	UPDATE NEW TRIP
Eiffel Tower		2	UPDATE NEW TRIP
Forbidden City		1	UPDATE NEW TRIP
Glenfinnan Viaduct		0	UPDATE NEW TRIP
London Towers		0	UPDATE NEW TRIP

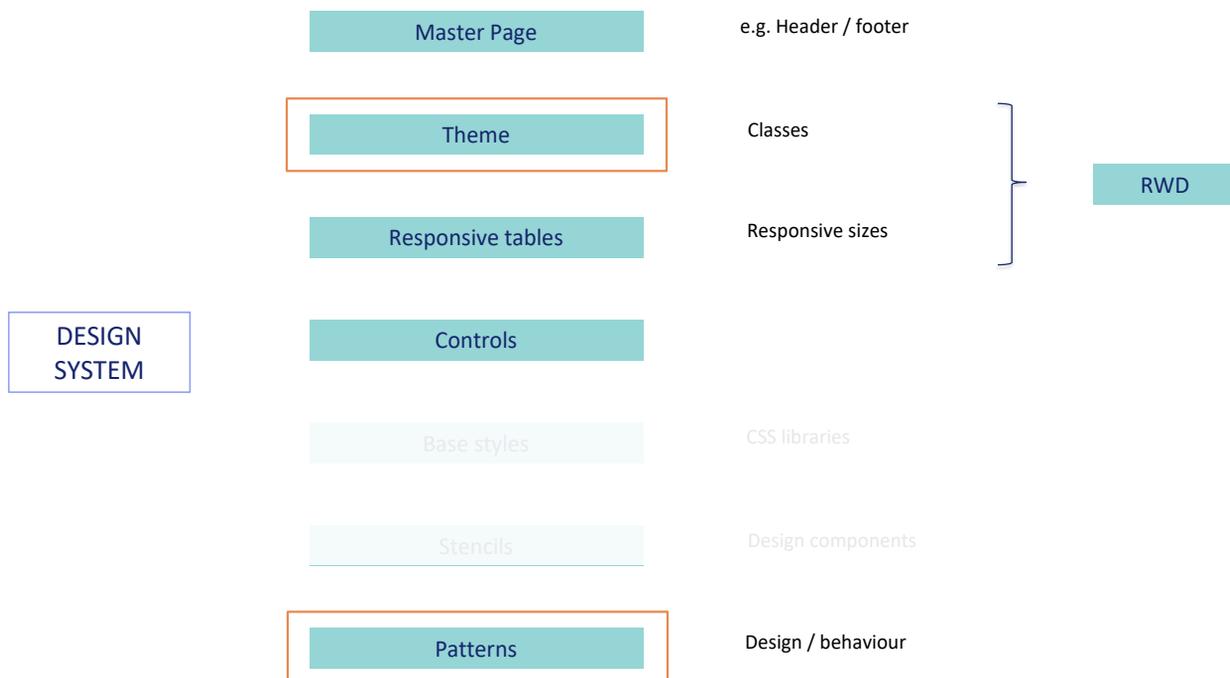


Column Class: WWOptionalColumn

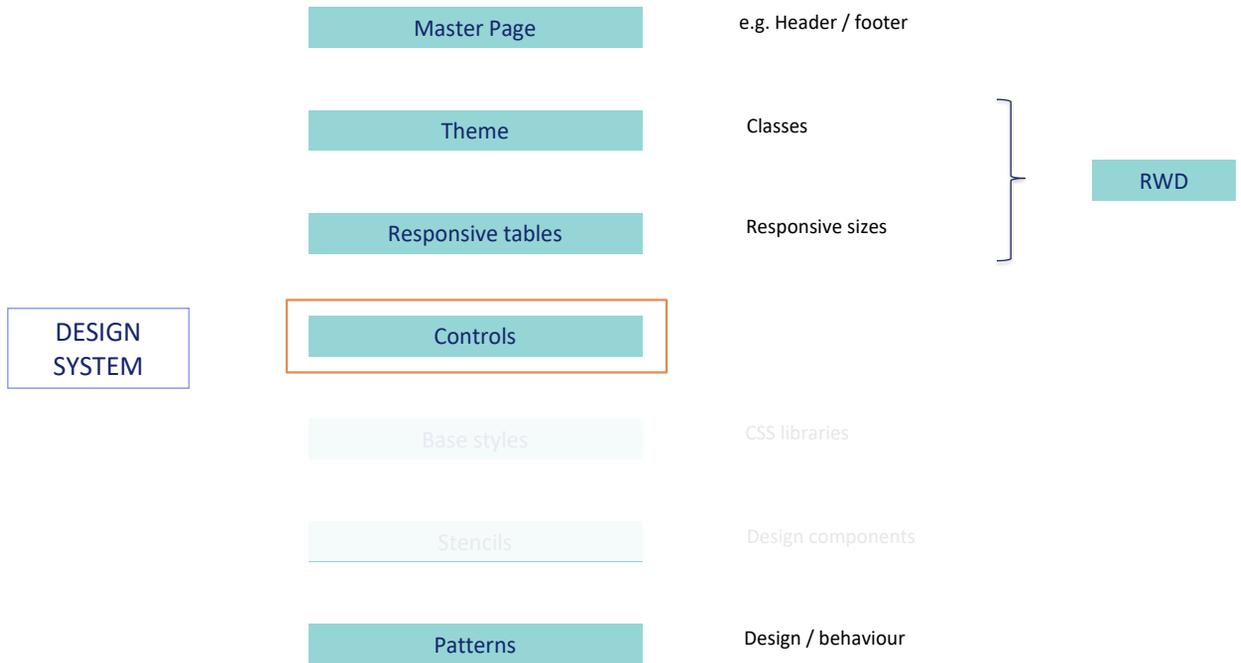
Então, se queremos conseguir para o grid de nosso web panel, que em tamanho Small não sejam exibidas outras colunas além das do nome de atração e ações, modificamos as classes das colunas de acordo com o que acabamos de ver.



Em resumo: outra parte importante do Design System tem a ver com a responsividade, que é obtida tanto variando as posições e a visibilidade dos controles das responsive tables, como variando as propriedades das classes de acordo também com o tamanho (através dessas colunas do Theme que vimos).

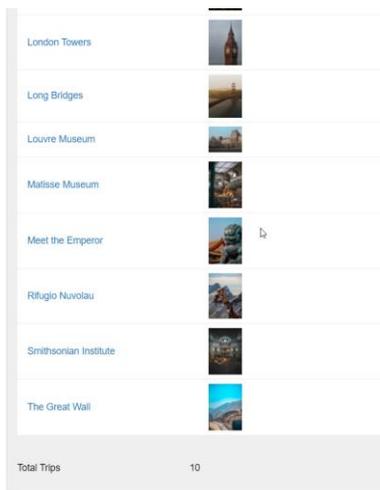


Para conseguir tudo isso, não tivemos que fazer [quase nada](#). Apenas imitar o que o pattern Work With fez tudo automaticamente, em conjunto com o theme. Por isso que dizemos que GeneXus já nos fornece um Design System básico, predefinido, que podemos utilizar para nossos web panels.

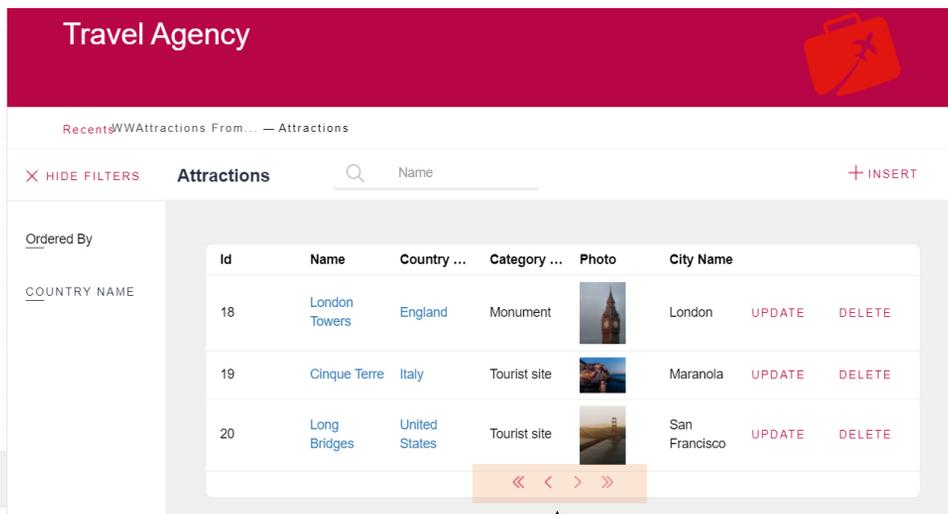


Outra parte de tudo isto terá a ver com os controlos que utilizemos.

Grids: design and behavior



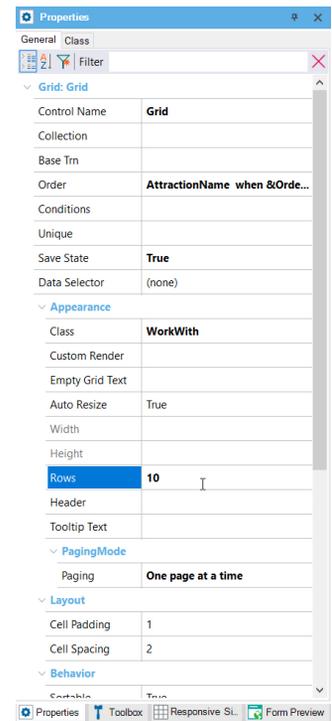
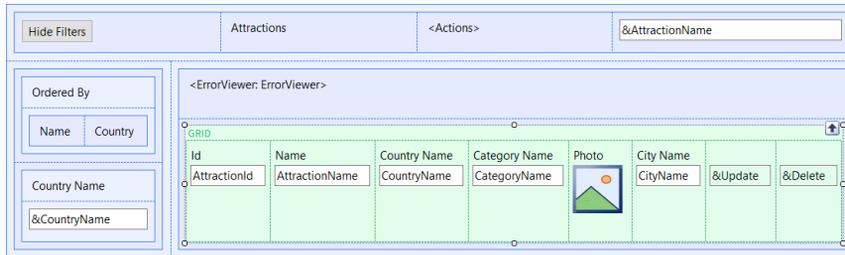
Our web panel: no grid paging



Work With web panel: grid paging

Por exemplo, em nosso web panel, onde imitávamos o do Work With, não temos a paginação que o Work With apresenta.

Grid Paging



Se observarmos as propriedades do grid, vemos a de nome Rows, que tem o valor 10, junto com a propriedade PagingMode que tem este valor. Fazemos o mesmo em nosso grid.

Grid Paging

Attraction Name From

Attraction Name To

Attraction Id	Attraction Name	Attraction Photo	Trips	Total Trips
AttractionId	AttractionName		&trips &update2 &newTrip	&totalTrips

Properties

General Class

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName w...
Conditions	CountryId = &CountryId whe...
Unique	
Save State	False
Data Selector	(none)

Appearance

Class	WorkWith
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	0
Header	
Tooltip Text	

Layout

Cell Padding	1
Cell Spacing	2

Behavior

Sortable	True
Allow Drop	False

Properties

General Class

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName w...
Conditions	CountryId = &CountryId whe...
Unique	
Save State	False
Data Selector	(none)

Appearance

Class	WorkWith
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	10
Header	
Tooltip Text	

PagingMode

Paging	One page at a time
--------	--------------------

Layout

Cell Padding	1
Cell Spacing	2

Behavior

Sortable	True
Allow Drop	False

O valor 0 indica que sejam carregadas todas as linhas. Quando alteramos seu valor para 10, estamos lhe dizendo que pague, o que significa: que traga 10 registros da base de dados. Vemos que aparece a propriedade Paging com o mesmo valor que tinha para o work with.

Grid Paging

Travel Agency



Recent VW Attractions From...

Country Id: (None) ▾

Attraction Name From:

Attraction Name To:

Attraction Name	Attraction Photo	Trips		
Rifugio Nuvolau		0	UPDATE	NEW TRIP
Smithsonian Institute		1	UPDATE	NEW TRIP
The Great Wall		0	UPDATE	NEW TRIP

« < > »

Total Trips: 1

Vamos gerar este objeto e testar.

Aqui vemos que haviam sido carregadas todas as atrações turísticas, mas se agora atualizamos, vemos a paginação em ação.

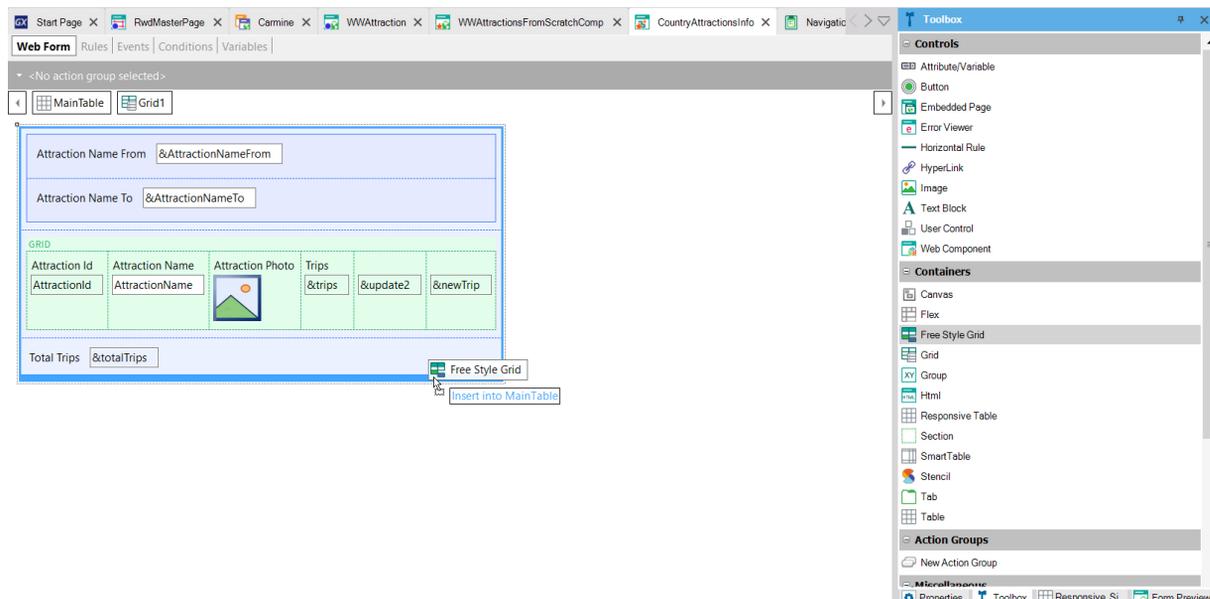
Grid Paging: infinite scrolling

The screenshot shows the GeneXus IDE interface. On the left, the Properties window for a Grid control is open. The 'PagingMode' property is set to 'Infinite scrolling'. The 'Rows' property is set to 5. On the right, the application preview is shown on a tablet. The application is titled 'Travel Agency' and displays a grid of attractions. The grid has 5 rows and 3 columns: 'Attraction Name', 'Attraction Photo', and 'Trips'. The attractions listed are Christ the Redemmer, Cinque Terre, Eiffel Tower, Forbidden City, and Glenfinnan Viaduct. A 'Loading...' message is visible at the bottom of the grid.

Attraction Name	Attraction Photo	Trips	UPDATE	NEW TRIP
Christ the Redemmer		3	UPDATE	NEW TRIP
Cinque Terre		0	UPDATE	NEW TRIP
Eiffel Tower		2	UPDATE	NEW TRIP
Forbidden City		1	UPDATE	NEW TRIP
Glenfinnan Viaduct		0	UPDATE	NEW TRIP

Agora vamos alterar o tamanho da página para que seja de 5 linhas, e modificamos a propriedade Paging para que assumo o valor Infinite Scrolling e vamos testar. Muda a visualização para um Tablet, para ver melhor. Vemos 5 linhas carregadas no grid, mas não temos os botões para ir para a próxima página. No entanto, ao fazer scroll, vemos a mensagem Loading. Estão sendo buscadas as próximas 5 linhas e são carregadas na tela e assim por diante.

Free style Grid



Por outro lado, as informações estão sendo apresentadas de forma estruturada, como se fosse uma tabela. E se quiséssemos ver as atrações turísticas de uma forma mais flexível?

Por exemplo, se quiséssemos ver apenas foto e nome, mas uma abaixo da outra? Para isso temos outro tipo de grid, o de estilo livre. Vamos arrastá-lo para o final do form. O inicializamos com os atributos AttractionName e AttractionPhoto. Os movemos para que fique a foto acima. E removemos a etiqueta.

Free style Grid

+ AttractionId



```

Event Start
...
AttractionId.Visible = False
...
endevent

```

Observemos que o grid tem as mesmas propriedades que o padrão para especificar transação base, Order, Conditions. Então, as copiamos do outro. Removemos esta variável agora, que não nos interessa. Omitimos adicionar AttractionId, atributo de que precisaremos, como no outro grid, que esteja carregado, embora o deixemos invisível. Isto era para poder passá-lo por parâmetro posteriormente.

Free style Grid

Cinque Terre		0	UPDATE	NEW TRIP
Eiffel Tower		2	UPDATE	NEW TRIP
Forbidden City		1	UPDATE	NEW TRIP
Glenfinnan Viaduct		0	UPDATE	NEW TRIP



Christ the Redemmer

Cinque Terre

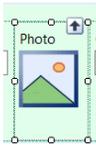
Eiffel Tower

Forbidden City

Bem, agora vamos testar. Atualizamos...

E vemos abaixo de nosso grid padrão, o novo, sem nenhum desenho, claro. Suponhamos que queremos que a imagem pareça muito maior. Por que está sendo exibida tão pequena?

Bigger image



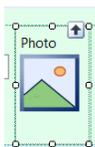
Properties	
General Class	
Attribute/Variable: AttractionPhoto	
Attribute	AttractionPhoto
Title	Photo
Class	ImageAttribute
Column Class	WWColumn WWOptionalColu...
Return On Click	False

The screenshot shows the GeneXus IDE interface. The main window displays a class hierarchy for the 'Photo' attribute. The 'ImageAttribute' class is highlighted, and its subclasses are listed: 'ActionAttribute', 'ProfileImageAttribute', 'ReadonlyImageAttribute', and 'ResponsiveImageAttrib...'. The 'ReadonlyImageAttribute' class is selected, and its properties are shown in the right-hand pane. The 'Width' property is set to 50px, and the 'Max-Width' property is set to 40px. The 'Properties' pane also shows other settings like 'Font', 'Text', and 'Text Shadow'.

Se observamos a classe do controle atributo no grid padrão e no freestyle, vemos que foi atribuída por padrão a mesma classe, ImageAttribute, porque se trata de um atributo de tipo de dados Image. Se formos ao theme para encontrá-la, veremos que possui uma série de subclasses, entre elas a de nome ReadonlyImageAttribute. Esta será a classe que será aplicada à nossa foto em execução, porque o controle atributo neste caso é readonly, então embora vejamos nas propriedades a classe pai, na verdade aplicará esta filha. Se olharmos suas propriedades, vemos que está especificando uma largura de 50 pixels para a imagem. E é por isso que a vemos tão pequena.

Se modificarmos aqui esta propriedade para que a largura seja, por exemplo de 400 px, conseguiremos o que queríamos para nosso grid, mas faremos, como efeito secundário, com que qualquer outro controle que tenha essa classe como sua, também será modificado. Em particular a imagem no Work With.

Bigger image



Properties

General Class

Filter

Attribute/Variable: **AttractionPhoto**

Attribute	AttractionPhoto
Return On Click	False
On Click Event	
Appearance	
Label Position	None
Class	ImageAttribute2
Invite Message	

Start Page x RwdMasterPage x Camine x WWAAttraction x WWAAttractionsFromScratchComp x Properties

Styles Colors Images

Filter

Order Rules

- AudioAttribute
- BlobContentAttribute
- BlobInputAttribute
- CheckBox
- CheckboxLabel
- ComboAttribute
- DescriptionAttribute
- DownloadAttribute
- ErrorAttribute
- FilterAttribute
- ImageAttribute
 - ActionAttribute
 - ProfileImageAttribute
 - ReadOnlyImageAttribute
 - ResponsiveImageAttrib...
- ImageAttribute2
 - BlobContentImageAttrib...
 - BlobInputImageAttribute2
 - ReadOnlyImageAttribu...**
- IME_Active
- IME_Disabled
- IME_Inactive

Output

Show: Build

Uploading 27 Kbytes

Deploying website

Success: Build With This Only

Properties

Filter

Top Right Radius

Bottom Left Radi

Bottom Right Rac

Font

FontStyle

FontVariant

FontWeight

FontSize

TextDecoration

FontFamily

Arial

Forecolor

IME Mode

Height

Max- Height

100%

Min- Height

Width

auto

Max- Width

400px

Min- Width

Mouse and Keyboard

Cursor

Text

Letter Spacing

Text Align

left

Text Indent

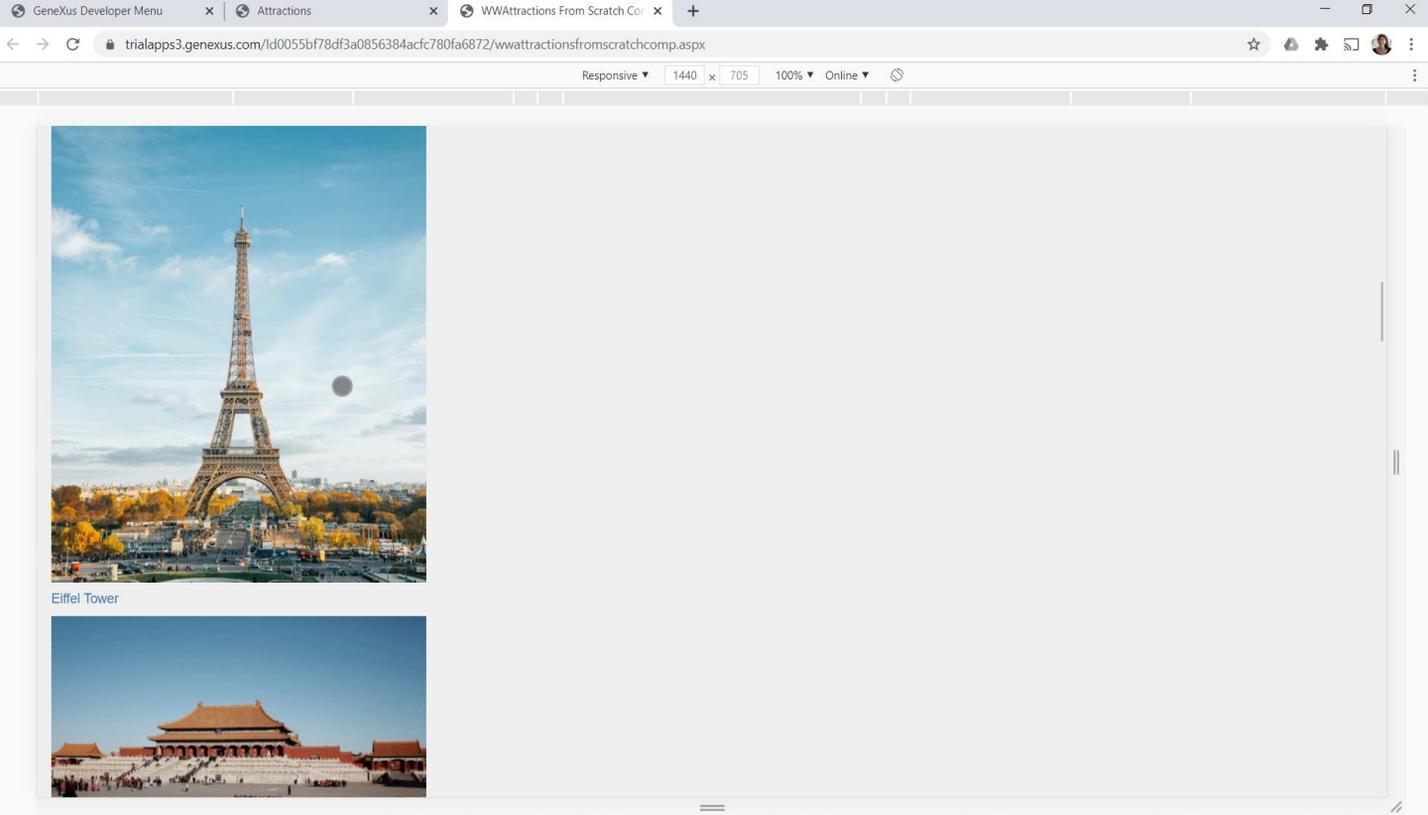
Text Transform

Vertical Align

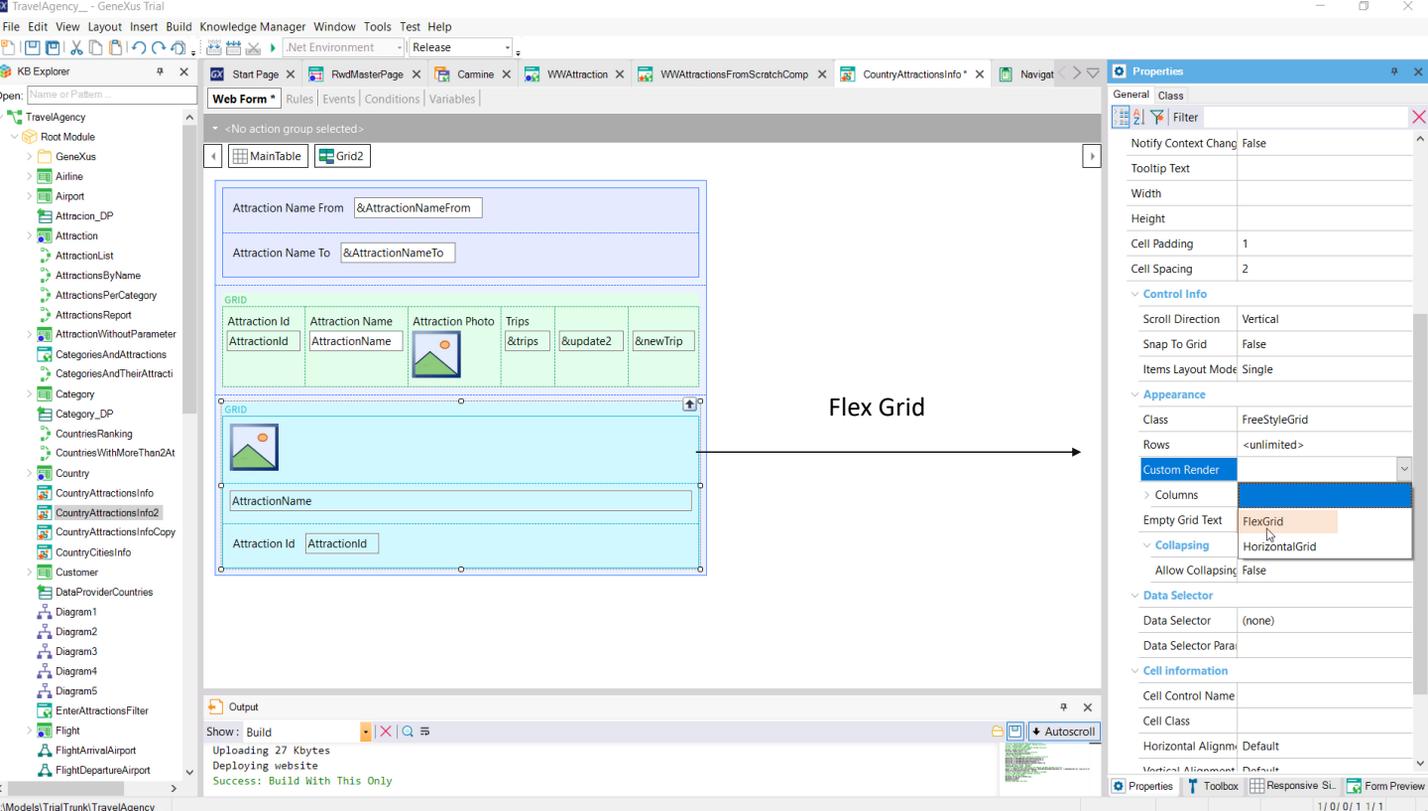
Properties | Toolbox | Responsive Sl. | Form Preview

Não queremos isto, então criamos outra classe irmã desta (já a mostramos criada), a que chamamos de ImageAttribute2, que foi criada automaticamente com todas estas subclasses, e o que fizemos foi modificar estas propriedades para a Readonly. Estamos estabelecendo que a imagem seja apresentada com seu tamanho original, desde que não exceda 400 px de largura. E, nesse caso, que seja diminuída até este tamanho.

Então, agora vamos modificar a classe do nosso controle AttractionPhoto para este segundo grid, para esta outra.



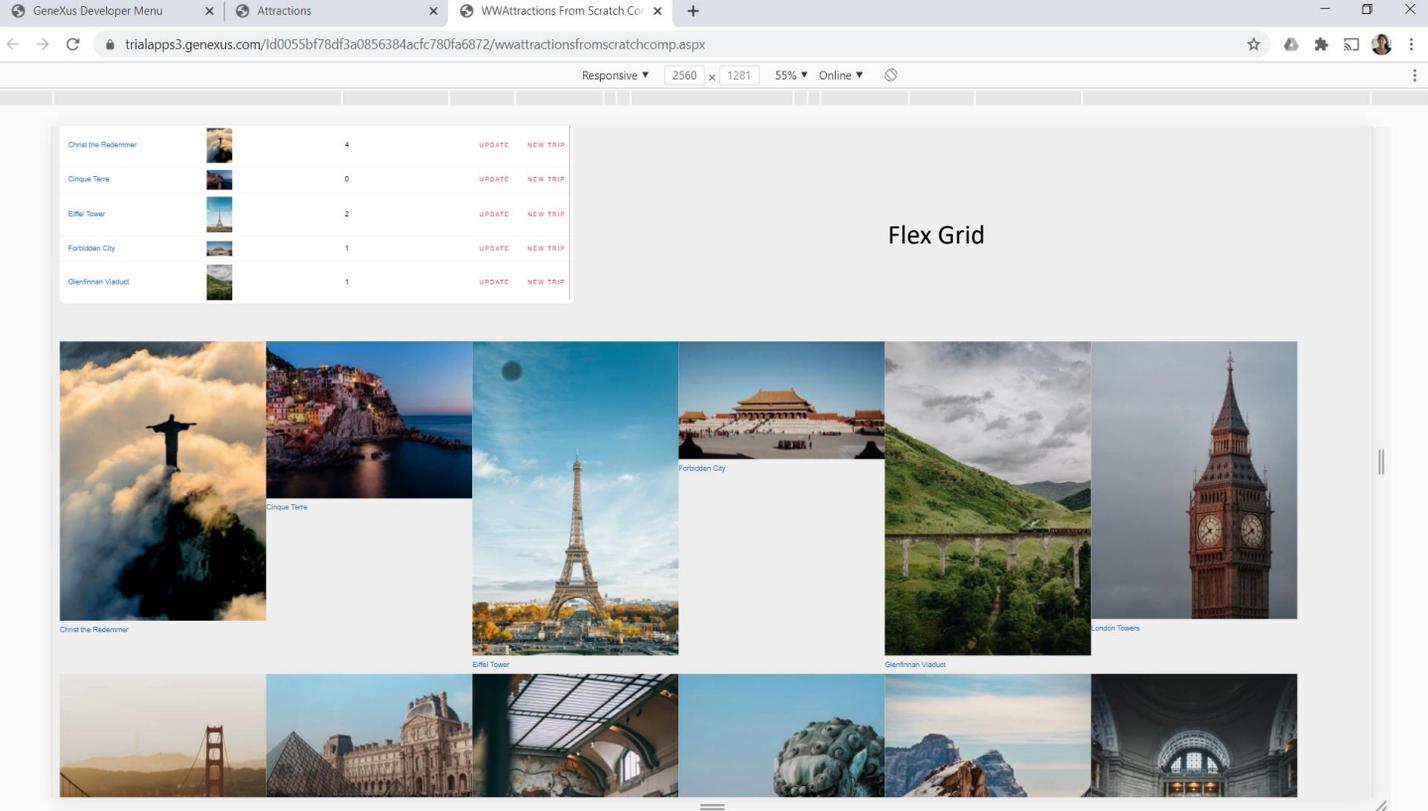
Testemos em execução. Se vê claramente a diferença.



E se agora quisermos apresentar as atrações de uma forma que aproveite mais o espaço da tela?

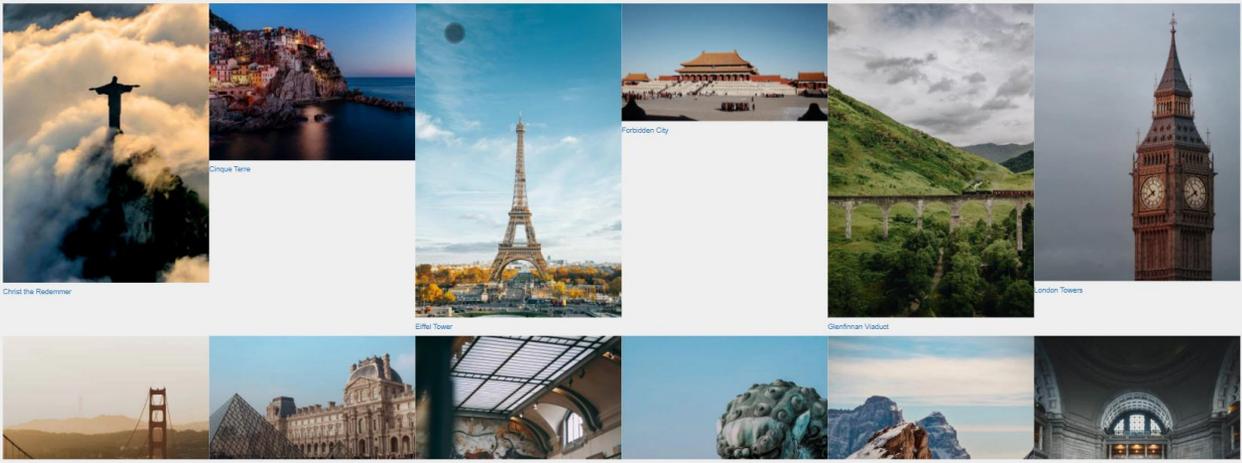
Vemos que as fotos não têm um tamanho homogêneo.

Vamos ao grid e modifiquemos seu rendering para que seja um grid flexível, ou seja, para que seu conteúdo ocupe como um quebra-cabeça os espaços livres da tela. Testemos.



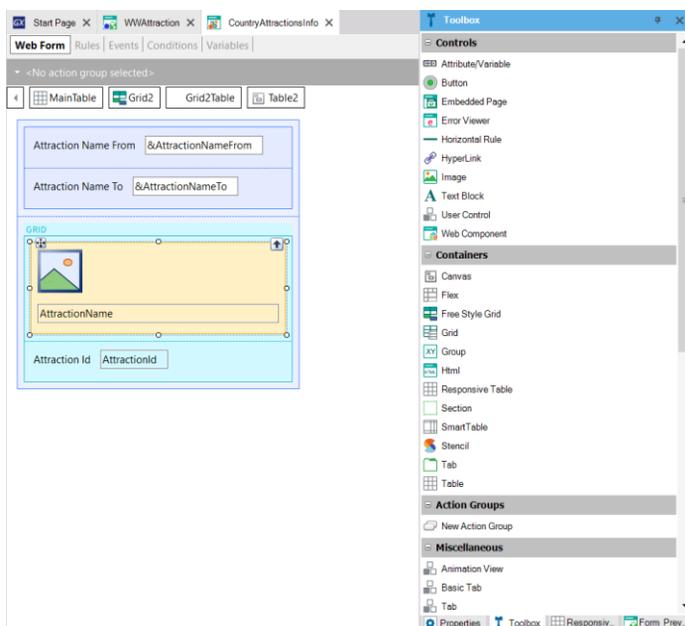
Christ the Redeemer		4	UPDATE	NEW TRIP
Cinque Terre		0	UPDATE	NEW TRIP
Eiffel Tower		2	UPDATE	NEW TRIP
Forbidden City		1	UPDATE	NEW TRIP
Glenfinnan Viaduct		1	UPDATE	NEW TRIP

Flex Grid

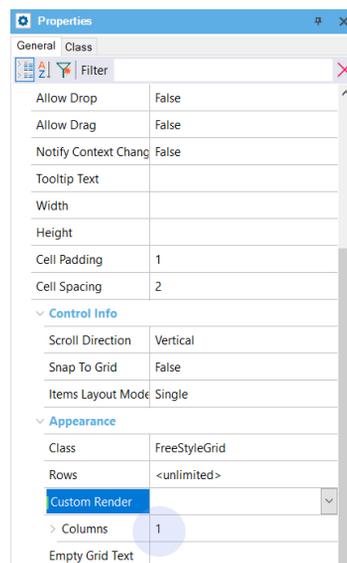


Vemos que dependendo do tamanho da tela, e considerando essa largura máxima de 400 pixels, as atrações turísticas vão se acomodando no sentido Row para preencher o espaço.

Canvas to overlapping



Default Free Style Grid

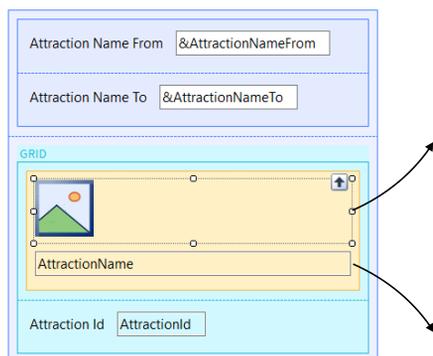


E se quiséssemos que o nome da atração apareça sobreposto na imagem?

Começamos eliminando o grid padrão que tínhamos. Não o mostramos, mas acabamos de eliminar também os eventos associados.

Agora inserimos um controle Canvas no form, que é um tipo de tabela que permite a sobreposição dos controles que contém. Movemos ali para dentro os dois controles que queremos que se sobreponham. Voltamos a deixar o grid freestyle com o rendering default, pelo qual será listada uma atração (com sua foto e nome) por linha.

Canvas to overlapping



Absolute position	
Top	0px
Left	0px
Bottom	0px
Right	0px
Width	100%
Height	100%
Z- Order	1

Absolute position	
Top	0px
Left	15px
Bottom	0px
Right	15px
Width	100%
Height	100%
Z- Order	2

Properties

General Class

Filter

Canvas: Table2

Control Name	Table2
Tooltip Text	
Header	
Appearance	
Width	100%
Height	300px
Class	Table
Cell information	
Cell Control Name	
Cell Class	
Horizontal Alignment	Default
Vertical Alignment	Default
Row information	
Row Height	
Row Class	

Vemos que ao ter colocado este controle dentro do Canvas, apareceram estas propriedades que são as que permitem localizar o controle em relação à tabela, dizer quanto ocupará e em que camada estará. Deixamos todos os valores padrão, exceto o da camada. Colocamos 1, porque queremos que a imagem fique na camada mais profunda, debaixo da próxima, aquela com o nome. O nome da atração vamos colocar na camada 2, em cima da 1. Queremos também que este controle tenha uma margem esquerda de 15 pixels, para que tenha uma margem em relação à foto.

Vejam que entre as propriedades da tabela Canvas, temos aquelas que nos permitem especificar quanto de altura e de largura ocupará em relação ao controle que a contém. Aqui estamos deixando que ocupe 100% da largura, mas vamos indicar que a altura seja de, por exemplo, 300 pixels.

The screenshot shows a web application development environment. On the left, there is a vertical list of attraction cards: 'Cinque Terre', 'Eiffel Tower', and 'Forbidden City'. The 'Eiffel Tower' card is currently selected. In the center, a 'Grid' component is visible, containing a preview of the selected card. The grid has two columns: 'Attraction Name From' and 'Attraction Name To'. Below the grid, there are two input fields: 'AttractionName' and 'AttractionId'. On the right, the 'Properties' panel is open, showing the 'Attribute TextOverImage' class for the 'AttractionName' field. The properties include: Border Width (0px), Border Radius (none), Font (20px Calibri), Font Style (normal), Font Weight (400), Text Decoration (none), and Font Family (Calibri). The background color is set to #FFF.

E, por último, mudemos o desenho do controle `AttractionName`, para torná-lo mais bonito.

Vemos que por padrão tem associada a classe `Attribute`, mas vamos alterá-la para uma que criamos no theme, com este nome. Se observamos suas propriedades, vemos que, por exemplo, especificamos para a cor da fonte um branco, a fonte Calibri, um tamanho de 20 pixels, etc.

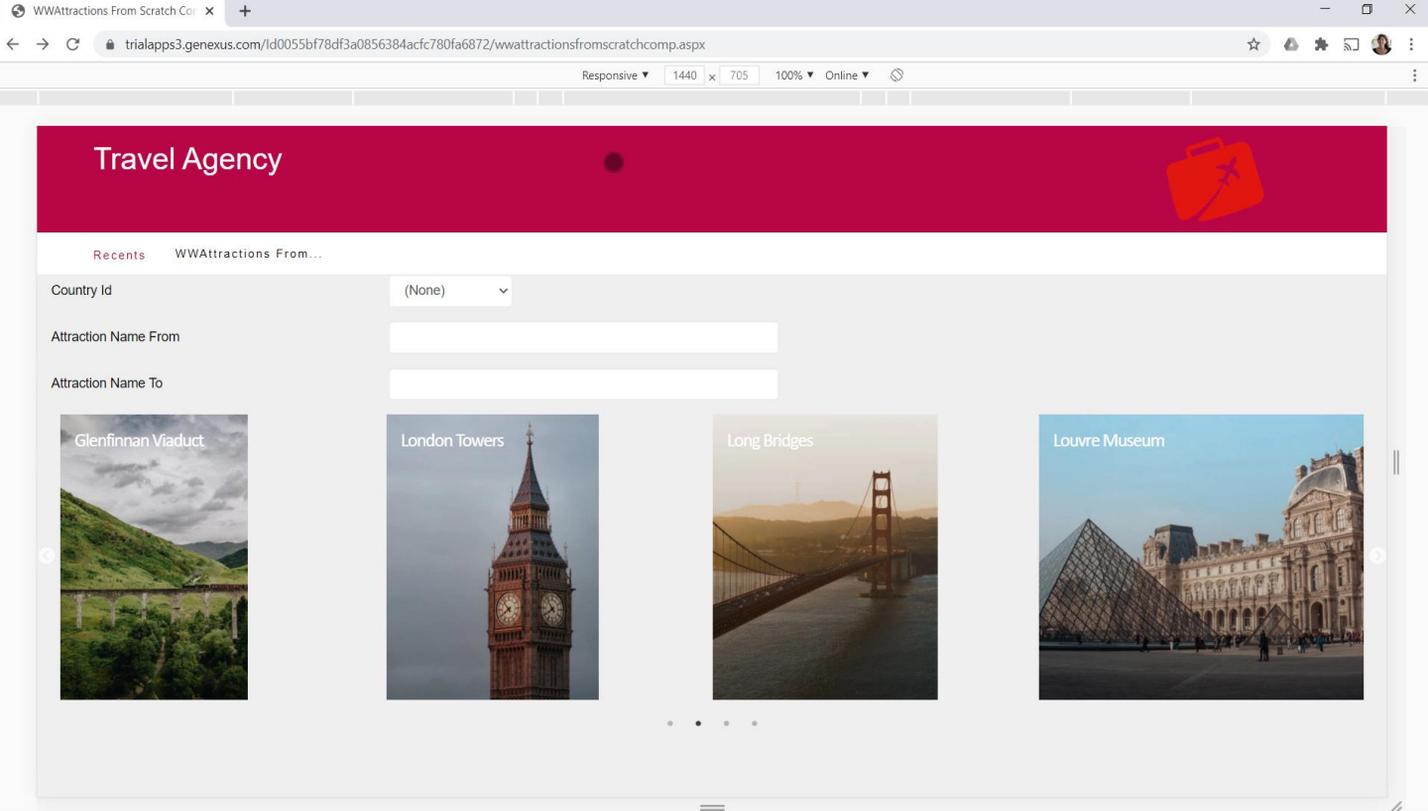
Vamos testar tudo isto em execução. Se agora clicamos no nome da torre Eiffel... vemos suas informações.

Horizontal grid

The screenshot shows the GeneXus IDE interface. The main window displays a web form with a grid control. The grid is currently rendered as a vertical grid. The properties panel on the right shows the following settings:

Property	Value
Cell Spacing	2
Control Info	
Scroll Direction	Vertical
Snap To Grid	False
Items Layout Mode	Single
Appearance	
Class	FreeStyleGrid
Rows	<unlimited>
Custom Render	HorizontalGrid
Columns	1 2 3 4
Extra Small	1
Small	2
Medium	3
Large	4
Empty Grid Text	

E para terminar de apreciar algumas das possibilidades que nos oferecem os controles grid, se agora modificamos seu rendering para que seja apresentado como um grid horizontal... onde, por exemplo, para tamanho tipo telefone queremos que mostre apenas uma coluna, para tamanho small, 2, para medium 3 e para large 4...



Pressionamos F12... Aqui vemos tamanho Large, com 4 atrações por página horizontal...

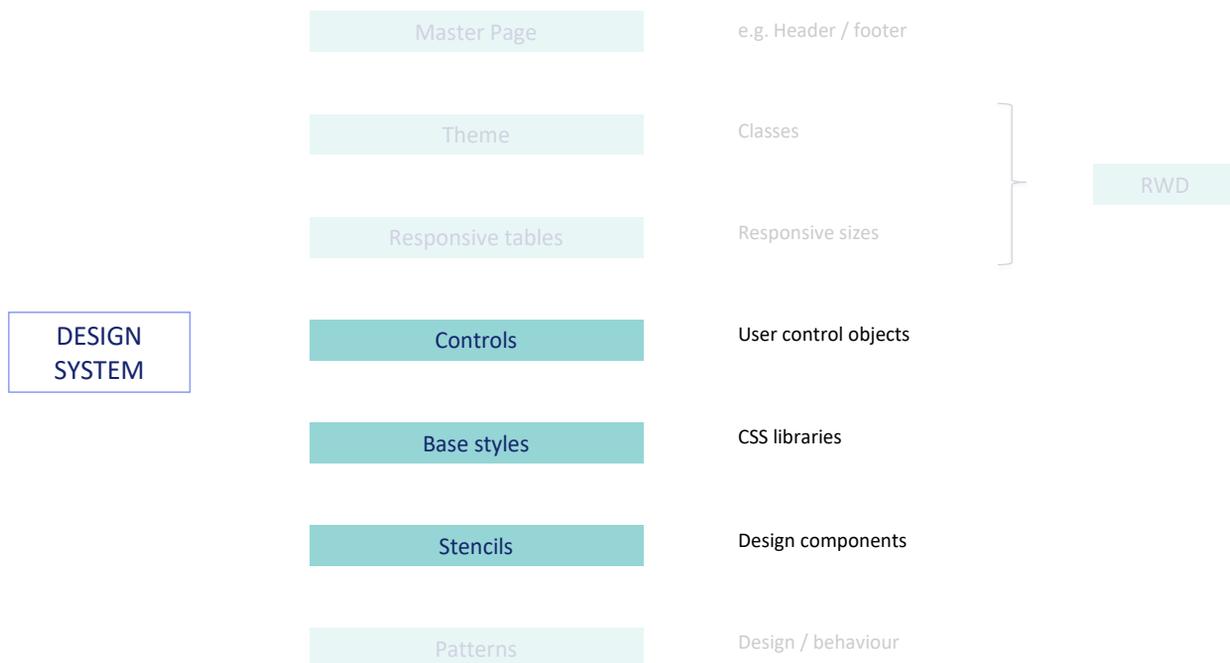
Aqui vemos 3...

Para um Tablet 2...

E para um telefone, uma.

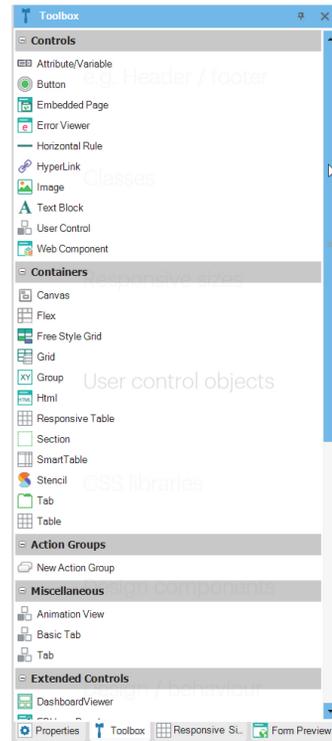
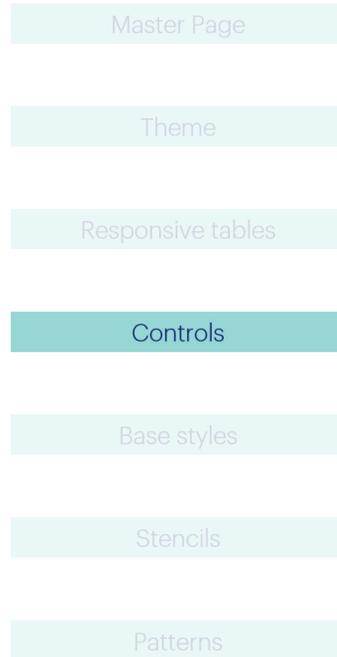
Claro, aqui será conveniente inserir imagens que tenham o mesmo tamanho.

Com isto, vimos apenas uma introdução a tudo o que podemos fazer com controles e as classes do theme.



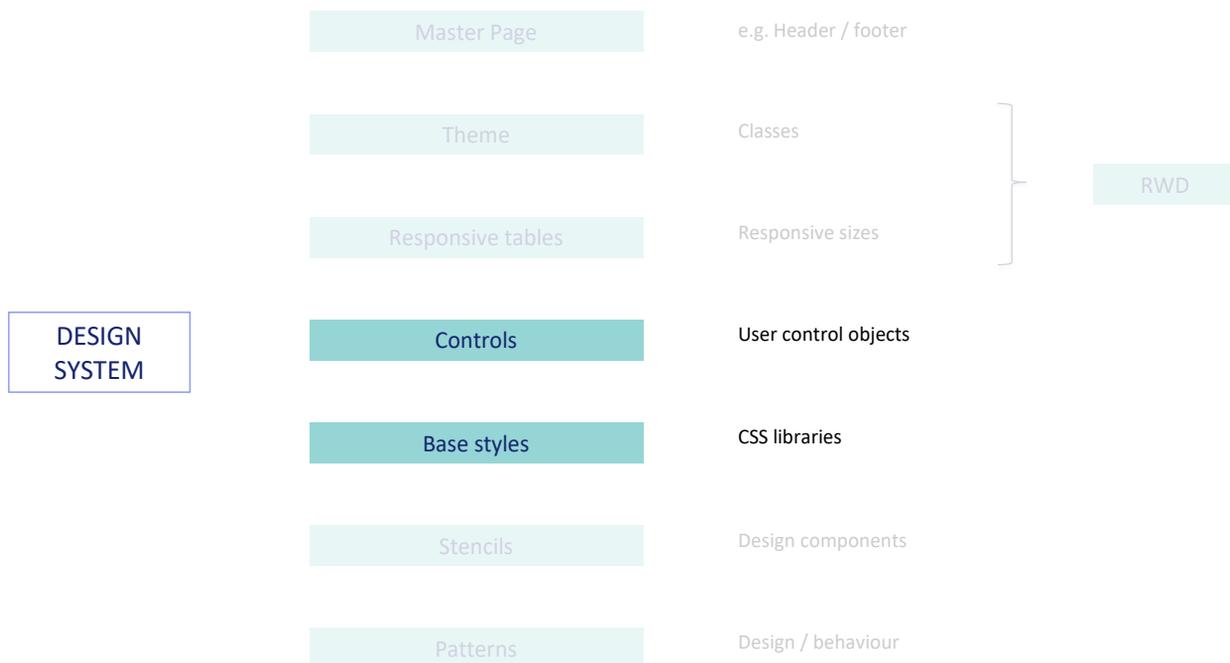
Existem mais alguns atores que, juntamente com todos os vistos, permitem utilizar no GeneXus um Design System potente. Não vamos estudá-los neste nível, mas os deixamos apresentados.

DESIGN
SYSTEM



RWD

Podemos não apenas utilizar em nossos forms os controles que vêm na toolbox do GeneXus de forma predeterminada...



...mas também podemos definir controles de usuário, que podem ser copiados de plataformas que oferecem esses controles junto com bibliotecas CSS (que são, como nossos themes, aquelas que especificam seu estilo –o estilo desses controles).

Card | Semantic UI

semantic-ui.com/views/card.html

UI Docs

Getting Started

New in 2.4

Introduction

- Integrations
- Build Tools
- Recipes
- Glossary

Usage

- Theming
- Layouts

Globals

- Reset
- Site

Elements

- Button
- Container
- Divider
- Flag
- Header
- Icon
- Image
- Input
- Label
- List
- Loader
- Placeholder

Card

A single card.

To ensure cards are equal height use the plural, `cards`. Card groups automatically uses `flex` styles to match height between cards in the same row.



Kristy
Joined in 2013
Kristy is an art director living in New York.
22 Friends

Por exemplo, esta, Semantic UI. Podemos querer usar um controle Card como este, por exemplo.

Para isso, conseguirá criando um objeto User control (podemos chamá-lo igual)...

Card | Semantic UI

semantic-ui.com/views/card.html

Card

A single card.

To ensure cards are equal height use the plural, `cards`. Card groups automatically uses flex styles to match height between cards in the same row.

Example



```
1 <div class="ui card">
2   <div class="image">
3     
5   <div class="content">
6     <a class="header">{{Name}}</a>
7     <div class="meta">
8       <span class="date">{{Date}}</span>
9     </div>
10    <div class="description">
11      {{Description}}
12    </div>
13  </div>
14  <div class="extra content">
15    <a>
16      <i class="user icon"></i>
17      {{ExtraContent}}
18    </a>
19  </div>
20 </div>
```

<div class="ui card">

...copiar e colar seu html, alterar os dados fixos por algo como nomes de elementos de um SDT (para torná-lo dinâmico, ou seja, ser capaz de carregar dinamicamente esse controle, com dados que poderemos especificar, por exemplo, da base de dados)...

Card | Semantic UI

semantic-ui.com/views/card.html

UI Docs

- Getting Started
- New in 2.4
- Introduction
 - Integrations
 - Build Tools
 - Recipes
 - Glossary
- Usage
 - Theming
 - Layouts
- Globals
 - Reset
 - Site
- Elements
 - Button
 - Container
 - Divider
 - Flag
 - Header
 - Icon
 - Image
 - Input
 - Label
 - List
 - Loader
 - Placeholder

Kristy
Joined in 2013
Kristy is an art director living in New York.
22 Friends

```
<div class="ui card">
  <div class="image">
    
  </div>
  <div class="content">
    <a class="header">Kristy</a>
    <div class="meta">
      <span class="date">Joined in 2013</span>
    </div>
    <div class="description">
      Kristy is an art director living in New York.
    </div>
    <div class="extra content">
      <a>
        <i class="user icon"></i>
        22 Friends
      </a>
    </div>
  </div>
</div>
```

Screen Template Properties

```
1 <div class="ui card">
2   <div class="image">
3     
5   <div class="content">
6     <a class="header">{{Name}}</a>
7     <div class="meta">
8       <span class="date">{{Date}}</span>
9     </div>
10    <div class="description">
11      {{Description}}
12    </div>
13  </div>
14  <div class="extra content">
15    <a>
16      <i class="user icon"></i>
17      {{extracontent}}
18    </a>
19  </div>
20 </div>
```

Properties

User Control: Card	
Name	Card
Description	Card
Module/Folder	Root Module
Is Control Type	False
References	
Base Control Type	None
Base Style	SemanticUI
Qualified Name	Card
Object Visibility	Public

...especificar de onde tomará o CSS, ou seja, o design das classes, fazer alguma coisa mais...

Card | Semantic UI

semantic-ui.com/views/card.html

UI Docs

Getting Started

New in 2.4

Introduction

Integrations

Build Tools

Recipes

Glossary

Usage

Theming

Layouts

Globals

Reset

Site

Elements

Button

Container

Divider

Flag

Header

Icon

Image

Input

Label

List

Loader

Placeholder

Kristy
Joined in 2013
Kristy is an art director living in New York.
22 Friends

```
<div class="ui card">
  <div class="image">
    
  </div>
  <div class="content">
    <a class="header">Kristy</a>
    <div class="meta">
      <span class="date">Joined in 2013</span>
    </div>
    <div class="description">
      Kristy is an art director living in New York.
    </div>
  </div>
  <div class="extra content">
    <a>
      <i class="user icon"></i>
      22 Friends
    </a>
  </div>
</div>
```

Web Form

Attraction Name From: &AttractionNameFrom

Attraction Name To: &AttractionNameTo

<Card: Card?>

Attraction Name

Attraction Id: &AttractionId

Controls

- Attribute/Variable
- Button
- Embedded Page
- Error Viewer
- Horizontal Rule
- HyperLink
- Image
- Text Block
- User Control
- Web Component

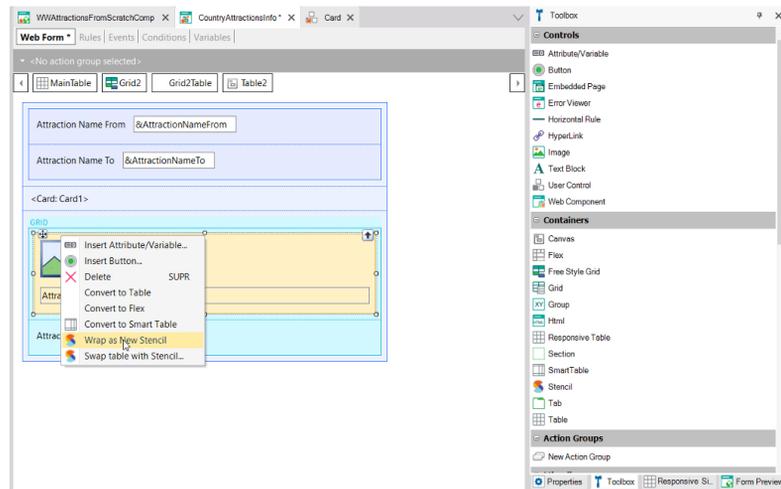
Containers

- Canvas
- Flex
- Free Style Grid
- Grid
- Group
- Html
- Responsive Table
- Section
- Smart Table
- Stencil
- Tab
- Table

Action Groups

- New Action Group

...e já o teremos disponível na Toolbox para utilizá-lo.



RWD

DESIGN
SYSTEM

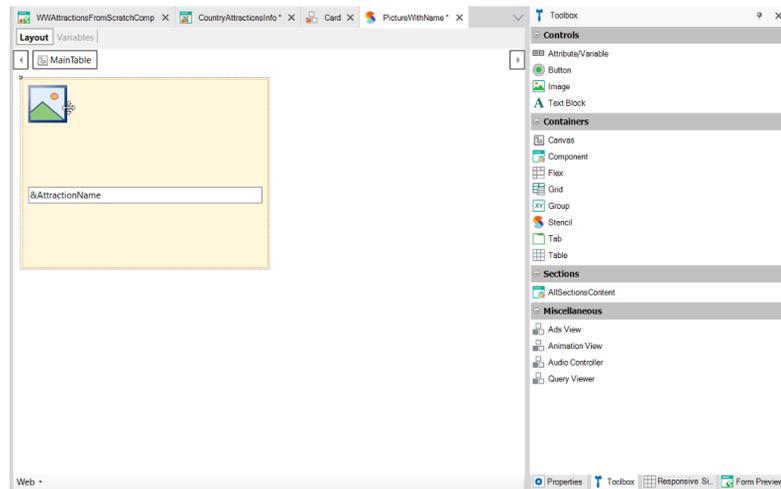
Stencils

Design components

Patterns

Design / behaviour

Por outro lado, GeneXus oferece Stencils, que servem para repetir o design de uma mesma parte de tela (um conjunto de controles), em muitas telas.



DESIGN
SYSTEM

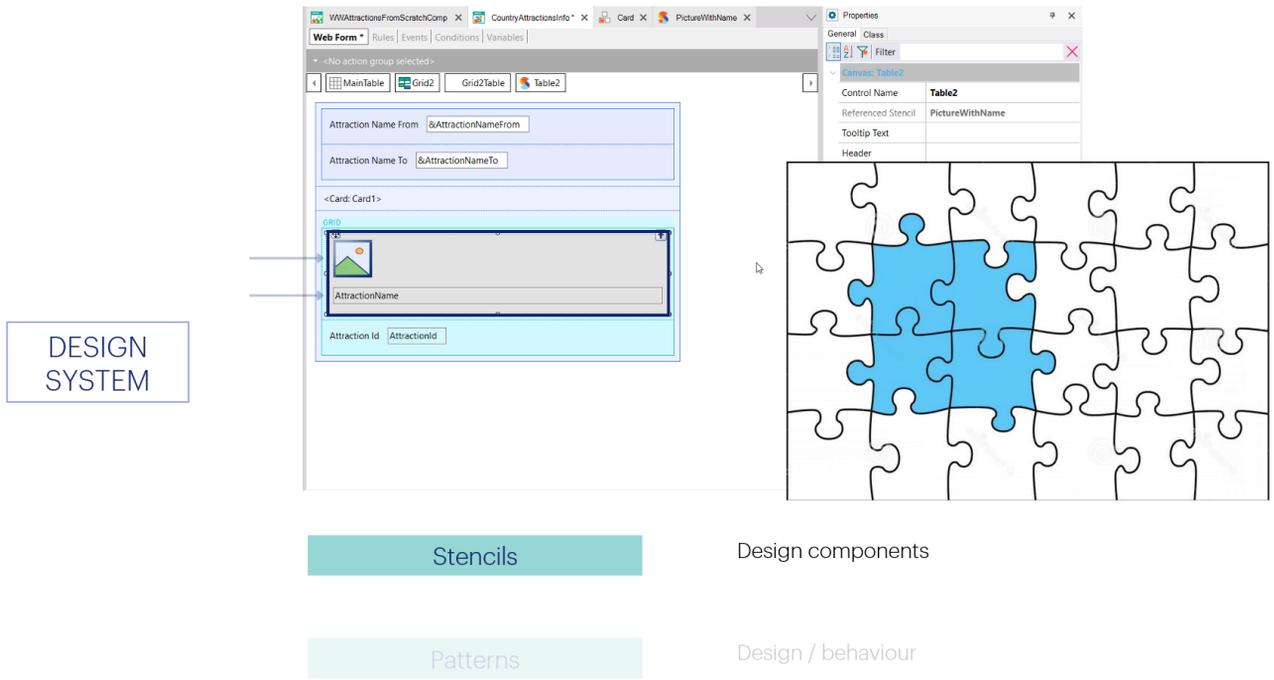
Stencils

Design components

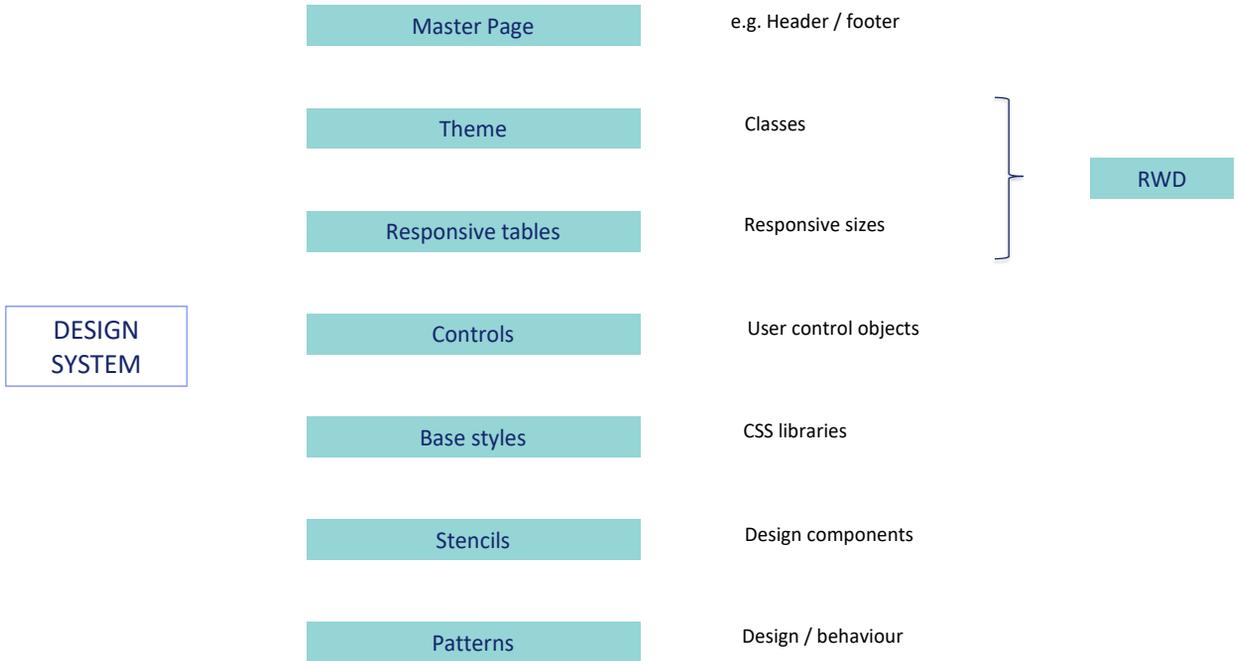
Patterns

Design / behaviour

É uma maneira de abstrair o design em um nível superior.



Se pudermos pensar nas telas de uma aplicação web como quebra-cabeças compostos de peças atômicas (que seriam os controles), os stencils são uma forma de agrupar um conjunto de controles cujo design se repetirá em várias telas. Seria uma peça feita de peças menores.



Com isto, terminamos de apresentar os atores mais importantes que fazem a implementação em GeneXus, do Design System.