

Design de uma aplicação Angular

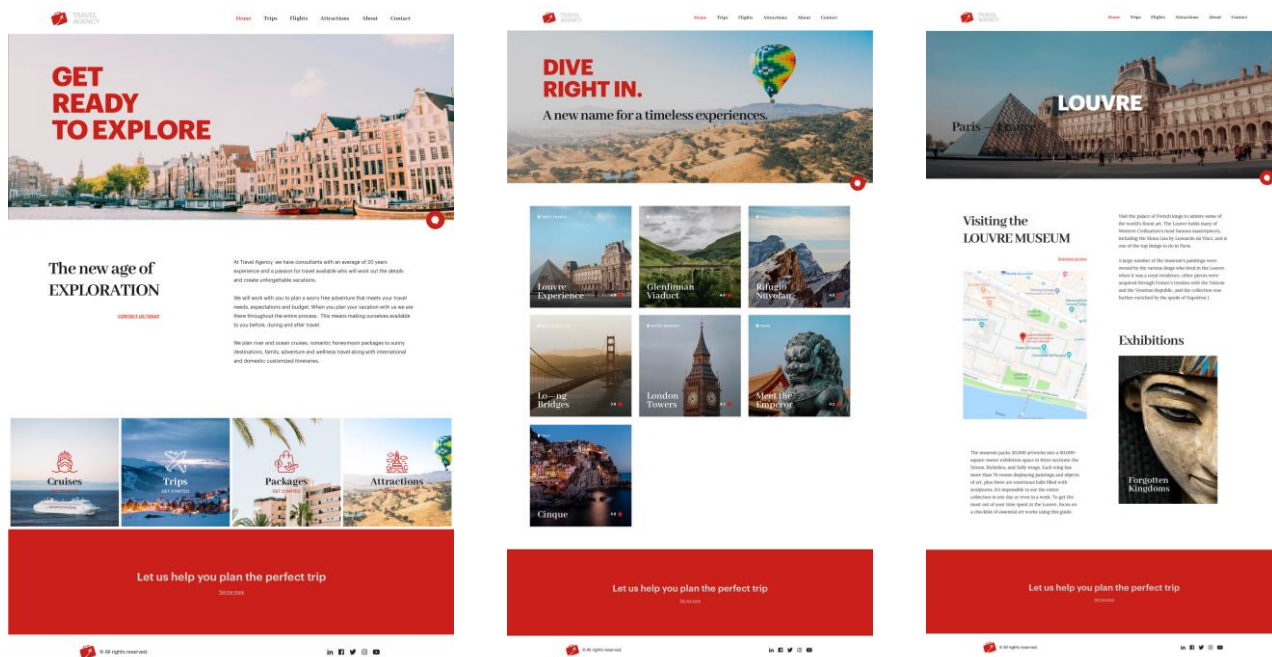
Importação de um design a partir do Sketch

GeneXus[™]

Em outro vídeo vimos como realizar nós mesmos as definições de design de nossa aplicação utilizando o Design System Object.

Neste vídeo veremos como podemos importar o design de telas completas, são especificações de cores, fontes e outras, feito por um profissional em design de aplicações digitais, em Sketch, sua ferramenta de trabalho habitual, e importar esse design em nossa aplicação GeneXus.

Design apresentado pelo designer



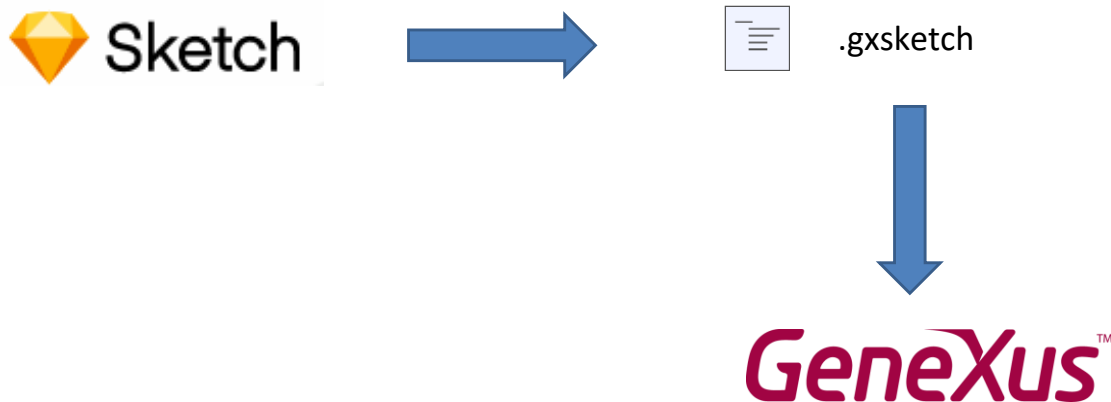
Combinamos com o designer que precisamos de uma tela inicial que nos sirva como ponto de partida para o restante das telas da aplicação, outra tela para ver a lista de atrações turísticas disponíveis e uma terceira tela de detalhe de uma atração turística.

Estas são as telas confeccionadas pelo designer. Na da esquerda, que é a tela principal, vemos acima de tudo o logotipo da aplicação e um menu de ações. Mais abaixo, uma imagem de fundo que tem sobre ela um título em destaque e abaixo à sua direita um botão para iniciar um chatbot, depois uma seção com dados da agência de viagens, abaixo dela quatro botões para acessar os dados das entidades que nossa aplicação irá gerenciar e abaixo deles um footer com um quadro vermelho com uma mensagem e os links para as redes sociais.

Em seguida, a tela do meio mostra na parte superior o logotipo, o menu, uma imagem contextual com título e botão de chatbot, um grid das atrações turísticas disponíveis e o footer. Vemos que muitos destes elementos também se repetem na terceira tela onde vemos o detalhe de uma atração, neste caso do Louvre.

Evidentemente o designer fez um bom trabalho, propondo um aspecto uniforme entre as telas, com uma forma clara e limpa de apresentar as informações.

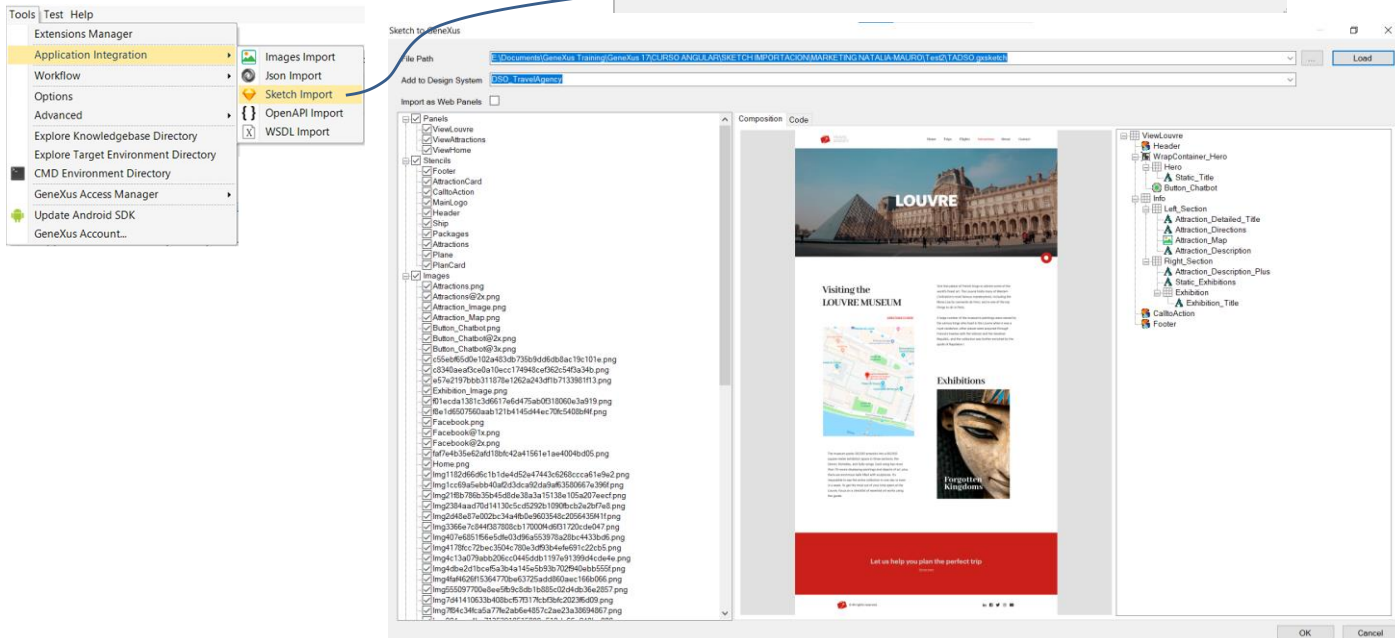
Importação a partir do Sketch



Este design foi feito na ferramenta que ele sempre usa, neste caso Sketch. Então utilizará um plugin de GeneXus instalado no Sketch que permite criar um arquivo de extensão gxsketch, que é o que nos envia.

GeneXus nos permite importar este arquivo criado na ferramenta do designer, diretamente em nossa KB e todos os elementos de design serão automaticamente criados em um Design System Object e inclusive serão criados os objetos GeneXus correspondentes com todos os controles necessários para que funcione a aplicação, com as telas que vimos antes.

Importação do arquivo .gxsksketch



Para importar o arquivo .sketch que o designer nos enviou, vamos para Tools/Application Integration e escolhemos Sketch import.

Adicionamos o nome do Design System Object onde queremos que fiquem salvas todas as definições de design.

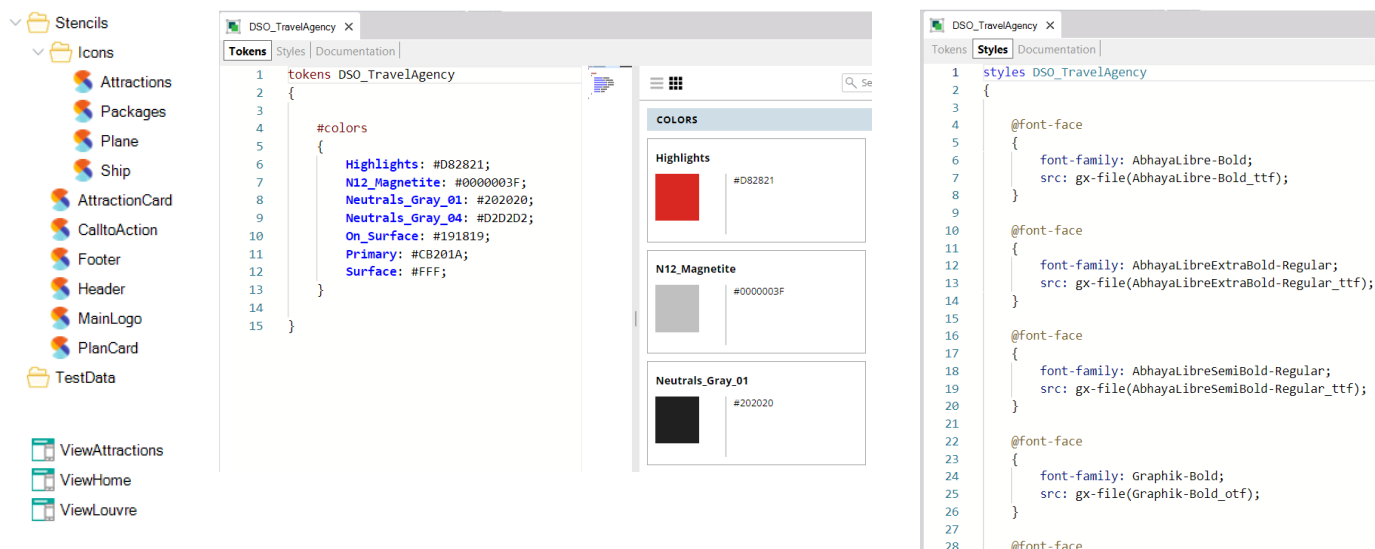
O check box Import as Web Panels deixamos desmarcado, porque queremos que sejam criados panels e não webpanels.

Se clicamos em algum dos panels que a importação criar, vemos sua pré-visualização e confirmamos que corresponde ao design que validamos. À direita, somos informados dos controles que conterà.

Vemos que também importará as imagens para poder executar o panel com dados fixos que nos permitirão executar e verificar se a funcionalidade é a desejada. Em seguida, deveremos substituir estes dados fixos pelos dados armazenados na base de dados de nossa aplicação.

Concordamos, então pressionamos OK.

Objetos criados na importação



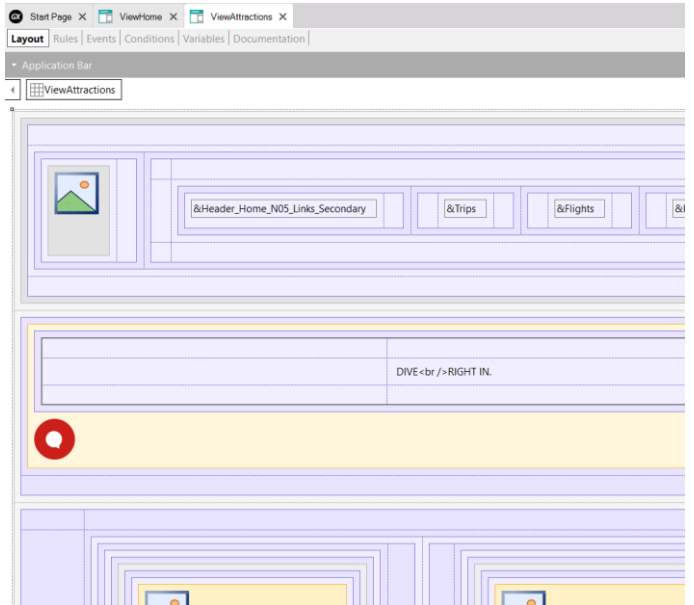
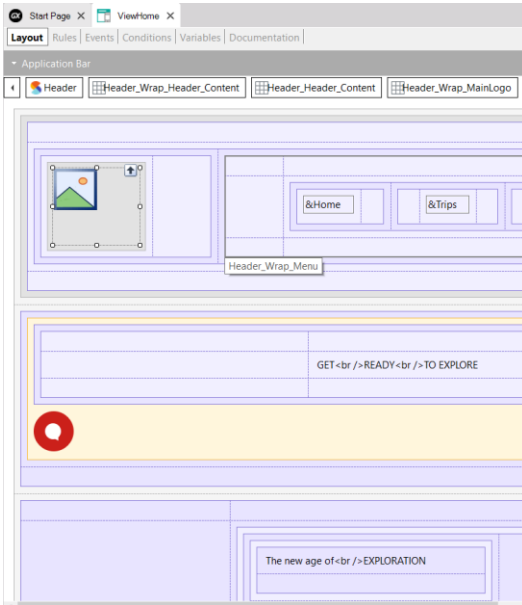
Na janela de Output são mostrados o progresso da importação e o resultado sem erros.

No KB Explorer, vemos que foram criadas duas pastas: Stencils, que contém vários stencils utilizados para encapsular design na aplicação, e TestData, que contém data providers e sdt para carregar dados fixos, o que nos permite testar a aplicação para ver se o design ficou como esperávamos.

Mais abaixo vemos que foi criado o Design System Object DSO_TravelAgency, que ao abri-lo vemos os Tokens e Styles que definem a aparência dos controles em tela.

Abaixo de tudo encontramos que foram criados automaticamente os objetos panels que vimos na tela do wizard de importação, como o da página inicial (ViewHome), o que mostra as atrações (ViewAttractions) e o detalhe de uma atração (ViewLouvre).

Objetos panels criados automaticamente na importação



Se vemos o form do ViewHome observamos que já estão incluídos todos os componentes para o conteúdo visual, como tabelas, variáveis, imagens, etc. tudo criado automaticamente no processo de importação. E o mesmo para o panel ViewAttractions.

Objetos criados na importação

The screenshot displays the GeneXus IDE interface. On the left, a small preview window shows a grid layout with a variable named '&Logo' containing an image of a green mountain and a sun. The main window shows the 'Variables' panel with a list of variables. The 'Logo' variable is selected, and its properties are shown in the 'Properties' panel on the right.

| Name | Type | Is Collection | Description |
|----------------------------|-----------------|-------------------------------------|---|
| Contact | VarChar(40) | <input type="checkbox"/> | Contact |
| Cruises | VarChar(40) | <input type="checkbox"/> | Cruises |
| Facebook | Image | <input type="checkbox"/> | Facebook |
| Flights | VarChar(40) | <input type="checkbox"/> | Flights |
| Home | VarChar(40) | <input type="checkbox"/> | Home |
| Information_Text | LongVarChar(2M) | <input type="checkbox"/> | Information_Text |
| Instagram | Image | <input type="checkbox"/> | Instagram |
| ISO1 | Image | <input type="checkbox"/> | ISO1 |
| Let_us_help_you_plan | VarChar(40) | <input type="checkbox"/> | Let_us_help_you_plan |
| link_information | VarChar(40) | <input type="checkbox"/> | link_information |
| LinkedIn | Image | <input type="checkbox"/> | LinkedIn |
| Logo | Image | <input checked="" type="checkbox"/> | Logo |
| Packages | Image | <input type="checkbox"/> | Packages |
| Plan_Image | Image | <input type="checkbox"/> | Plan_Image |
| PlanCardAttractions_Cr... | VarChar(40) | <input type="checkbox"/> | Plan Card Attractions_Cruises |
| PlanCardAttractions_Fla... | Image | <input type="checkbox"/> | Plan Card Attractions_Plan_Image |
| PlanCardCruises1 Cruis... | VarChar(40) | <input type="checkbox"/> | Plan Card Cruises1 Cruises N01 Titles H4 Negative |

| Variable: &Logo | |
|-----------------------|-------------------|
| Name | Logo |
| Description | Logo |
| Column title | Logo |
| Class | Image |
| Exo Gen Definition Ve | |
| Type Definition | |
| Based on | (none) |
| Data Type | Image |
| Collection | False |
| Initial value | Image:Logo.link() |

Se abrimos o panel ViewHome, e vamos ver as variáveis, vemos que as imagens foram carregadas mediante a propriedade Initial Value, fazendo referência às imagens que o processo de importação carregou na KB.

E se vamos aos eventos, vemos que também está programada a interação entre painéis, pois ao pressionar o botão de atrações invoca o panel ViewAttractions.

Objeto de startup de nossa aplicação

The screenshot displays the GeneXus IDE interface. On the left, a tree view shows the project structure under 'Menu', including 'Items' and three actions: 'Action (ViewAttractions)', 'Action (ViewHome)', and 'Action (ViewLouvre)'. The main editor shows the 'Events' tab for the 'Menu' object, containing the following code:

```

1  /* Generated by GeneXus Design Import [Start] */
2
3  Event 'ViewAttractions'
4    ViewAttractions()
5  EndEvent
6
7  Event 'ViewHome'
8    ViewHome()
9  EndEvent
10
11 Event 'ViewLouvre'
12   ViewLouvre()
13 EndEvent
14
15 /* Generated by GeneXus Design Import [End] */
16

```

On the right, the 'Properties' window is open for the 'Menu: TADSOMenu' object, showing the following details:

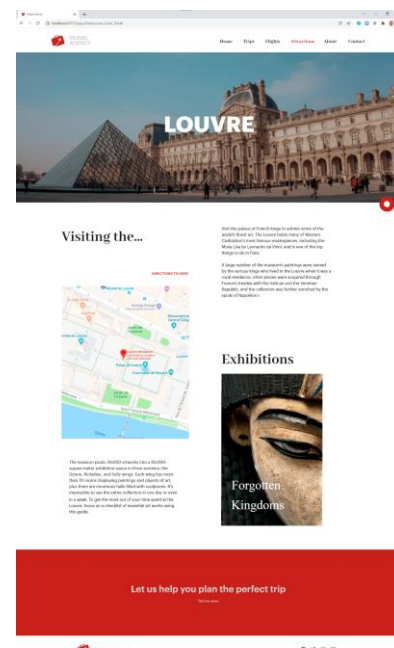
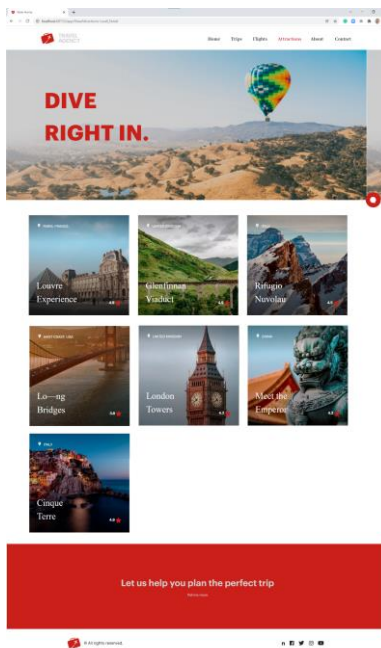
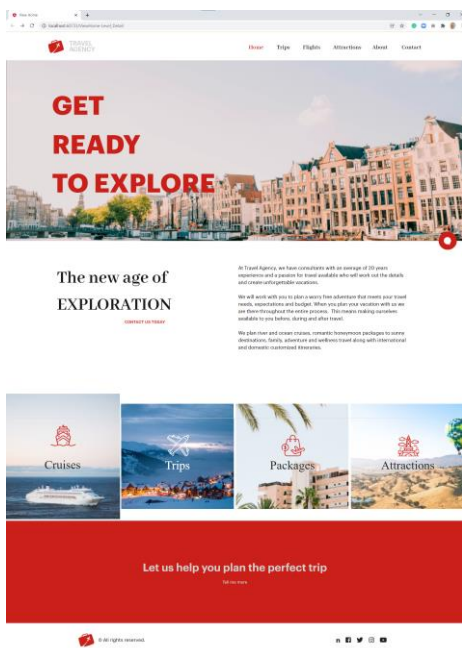
| | |
|---------------------------------------|-------------|
| Name | TADSOMenu |
| Description | TADSOMenu |
| Module/Folder | Root Module |
| Qualified Name | TADSOMenu |
| Object Visibility | Public |
| Automatic Update | False |
| Main program | True |
| Master Panel | (none) |
| Generate OpenAPI into Use Environment | |
| Miscellaneous | |
| Generate Object | True |
| Network | |
| Connectivity Support | Online |

Aqui podemos ver que também foi criado um objeto menu, que por padrão seria o objeto main a ser executado. Vemos que nos eventos, são invocados os outros painéis da aplicação.

Este objeto só será visto na geração para plataforma nativa, no caso de Angular, o objeto que será visto primeiro é o ViewHome, que é a segunda opção do menu.

Para executar, colocamos o objeto ViewHome como Startup Object e damos F5.

A aplicação em execução com dados fixos



Vemos que é executada a tela inicial e que ficou como previu nosso designer!.

Clicamos em Attractions e vemos a lista de atrações turísticas da agência de viagens.

Lembremos que estamos vendo os dados fixos que foram carregados durante a importação.

Se clicamos no Louvre, vemos que aparece uma página com informações detalhadas, tudo com um design limpo e esteticamente agradável.

Este exemplo permitiu-nos ver como é o processo de desenvolvimento de uma aplicação e como é importante trabalhar em equipe com pessoas de diferentes perfis, cada uma contribuindo para a melhor solução.

Modificação dos novos objetos para que acessem dados da base de dados

```

Layout | Rules | Events | Conditions | Variables | Documentation
Events
1  | /* Generated by GeneXus Design Import [Start] */
2  |
3  | Event Grid_Attractions.Load
4  |     Composite
5  |         For &ViewAttractions_Grid_Attractions_SDT in ViewAttractions_Grid_Attractions_DP()
6  |             Grid_Attractions.ItemLayout = &ViewAttractions_Grid_Attractions_SDT.Layout
7  |             load
8  |         EndFor
9  |     EndComposite
10 | EndEvent

```

| Name | Type |
|--------------------------------------|-------------|
| ViewAttractions_Grid_Attractions_SDT | |
| Layout | VarChar(40) |
| Attraction_Image | Image |
| Opacity_Adjustment_2 | Image |
| Attraction_Location | VarChar(40) |
| Location_Icon | Image |
| Opacity_Adjustment_1 | Image |
| Attraction_Name | VarChar(40) |
| Star_Icon | Image |
| Rating_Value | VarChar(40) |

```

Start Page X | ViewAttractions_Grid_Attractions_DP X
Source | Rules | Variables | Help | Documentation
1  | ViewAttractions_Grid_Attractions_DP
2  | {
3  |     ViewAttractions_Grid_Attractions_SDT
4  |     {
5  |         Layout = "AttractionItem4"
6  |         Attraction_Image = Image:e04d854f986edeff804aafef50f757e5120c94ce.Link()
7  |         Opacity_Adjustment_2 = Image:Opacity_Adjustment_2.Link()
8  |         Attraction_Location = Upper("Italy")
9  |         Location_Icon = Image:Location_Icon.Link()
10 |         Opacity_Adjustment_1 = Image:Opacity_Adjustment_1.Link()
11 |         Attraction_Name = "Rifugio" + NewLine() + "Nuvo!au"
12 |         Star_Icon = Image:Star_Icon.Link()
13 |         Rating_Value = "4.5"
14 |     }
15 |     ViewAttractions_Grid_Attractions_SDT
16 |     {
17 |         Layout = "AttractionItem4"
18 |         Attraction_Image = Image:b8d1a9828c1a76b0bd62713259d82fc5055db4c1.Link()
19 |         Opacity_Adjustment_2 = Image:Opacity_Adjustment_2.Link()
20 |         Attraction_Location = Upper("United Kingdom")
21 |         Location_Icon = Image:Location_Icon.Link()
22 |         Opacity_Adjustment_1 = Image:Opacity_Adjustment_1.Link()
23 |         Attraction_Name = "Glenfinnan" + NewLine() + "Viaduct"
24 |         Star_Icon = Image:Star_Icon.Link()
25 |         Rating_Value = "4.9"
26 |     }
27 | }
ViewAttractions_Grid_Attractions_SDT

```

Até agora, vimos dados fixos de teste, que os designers adicionaram para que possamos ver o comportamento da aplicação com o novo design. Então, vamos fazer algumas alterações, para que possamos ver os dados reais de nossa base de dados.

Se vamos para os eventos do panel ViewAttractions, vemos que o grid é carregado utilizando uma variável tipo SDT chamada &ViewAttractions_Grid_Attractions_SDT e o data provider ViewAttractions_Grid_Attractions_DP.

Vamos substituir a carga do DP para que em vez de pegar dados fixos, utilize os atributos da tabela Attraction e da tabela Country.

Modificação dos novos objetos para que acessem dados da base de dados

The image shows two screenshots from the GeneXus IDE. The top screenshot displays the source code for the data provider `ViewAttractions_Grid_Attractions_DP_BD`. The code defines a `ViewAttractions_Grid_Attractions_SDT` object with the following properties:

```

1 ViewAttractions_Grid_Attractions_DP_BD from Attraction
2 {
3   ViewAttractions_Grid_Attractions_SDT
4   {
5     Layout = "AttractionItem"
6     Attraction_Image = AttractionPhoto
7     Opacity_Adjustment_2 = Image:Opacity_Adjustment_2.Link()
8     Attraction_Location = CountryName
9     Opacity_Adjustment_1 = Image:Opacity_Adjustment_1.Link()
10    Attraction_Name = AttractionName
11    Star_Icon = Image:Star_Icon.Link()
12    Rating_Value = str(AttractionRating)
13  }
14 }
15

```

The bottom screenshot shows the `Events` tab for the `Grid_Attractions_Small` object. It contains the following event code:

```

1 /* Generated by GeneXus Sketch Import [Start] */
2
3 Event Grid_Attractions_Small.Load
4   For &ViewAttractions_Grid_Attractions_Small_SDT in ViewAttractions_Grid_Attractions_Small_DP_BD()
5     Grid_Attractions_Small.ItemLayout = &ViewAttractions_Grid_Attractions_Small_SDT.Layout
6     load
7   EndFor
8 EndEvent

```

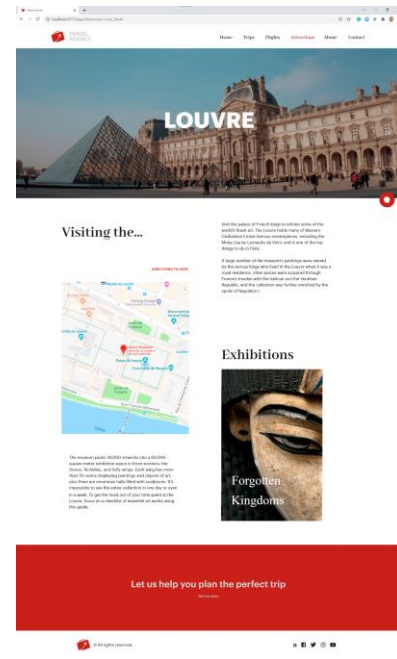
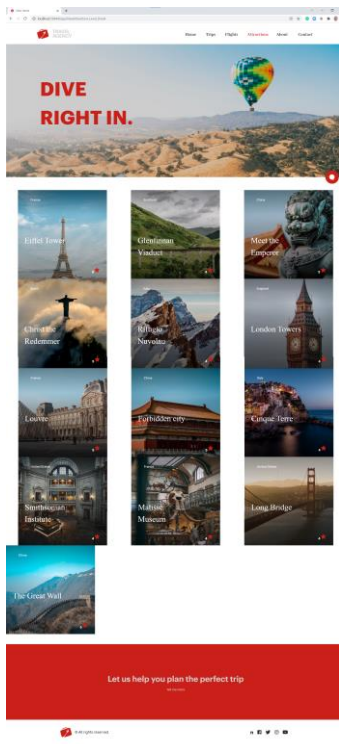
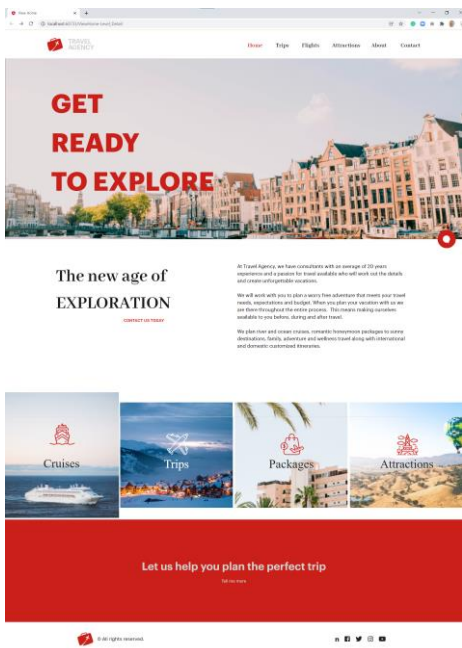
Fazemos Save As do Data Provider e salvamos o novo data provider com o nome `ViewAttractions_Grid_Attractions_DP_BD`.

Adicionamos a cláusula `from Attractions` e carregamos o campo `Attraction_image` com o valor do atributo `AttractionPhoto`, o campo `Attraction_name` com o atributo `AttractionName`, para `Attraction_location` atribuímos o atributo `CountryName` e para `Attraction_rating` atribuímos ao valor do atributo `AttractionRating`. Apagamos tudo o que não nos serve.

Agora voltamos ao panel `ViewAttractions` e alteramos o data provider que carrega as atrações, para o novo que construímos.

Salvamos, clicamos com o botão direito do mouse sobre o panel `ViewHome` e escolhemos `Run`.

Execução da aplicação com dados reais



Se vamos para as atrações, vemos que estamos percorrendo as atrações que tínhamos carregadas na base de dados, pois vemos a Torre Eiffel, ou o Viaduto Glennfinn que não estavam nos dados fixos e se paginamos vemos que estão todas as atrações. Se clicamos no Louvre, vemos a página do detalhe da atração.

Estamos cumprindo nosso objetivo solicitado pela Agência de mostrar todas as atrações turísticas, mas agora com a aplicação com um design mais adequado.

Não há dúvida de que se tivermos um designer em nossa equipe, nunca faríamos o trabalho de escrever à mão o DSO do zero, mas também a importação destas definições a partir do Sketch resolve o design de maneira mais profissional e podemos nos dedicar à parte funcional de nossa aplicação.

Esta forma de trabalhar em que integramos o design de um profissional em nosso projeto Genexus nos oferece muitas vantagens, pois é otimizado o esforço, fazendo com que cada integrante da equipe contribua de acordo com seu perfil, no que faz melhor.

Nos próximos vídeos, abordaremos outros temas, como adicionamos segurança à nossa aplicação ou como fazemos para colocá-la em produção nas instalações do cliente.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications