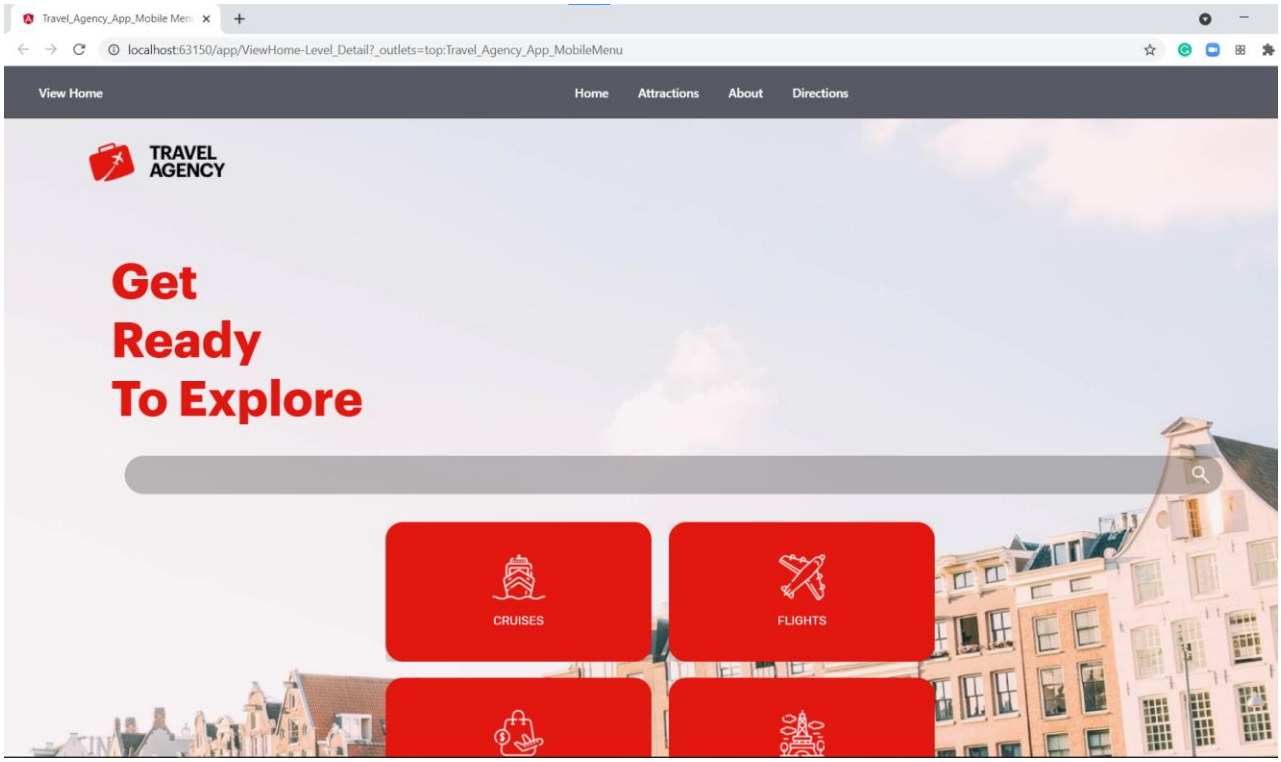
Primeiros passos com Angular

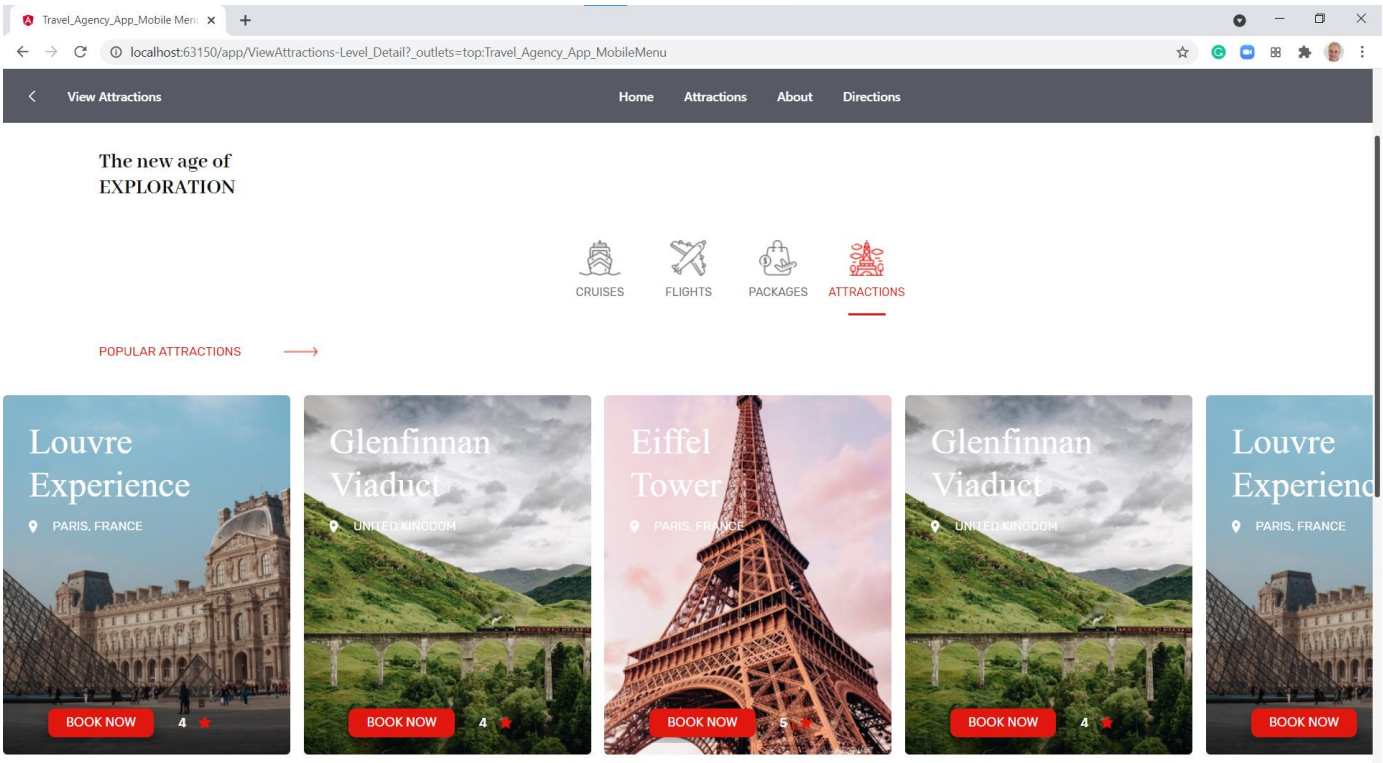
GeneXus™

No desenvolvimento do curso mostraremos como implementamos uma aplicação para os clientes de uma Agência de Viagens e faremos todos os nossos exemplos completando o seu desenvolvimento. Mas primeiro vejamos a aplicação já em funcionamento.

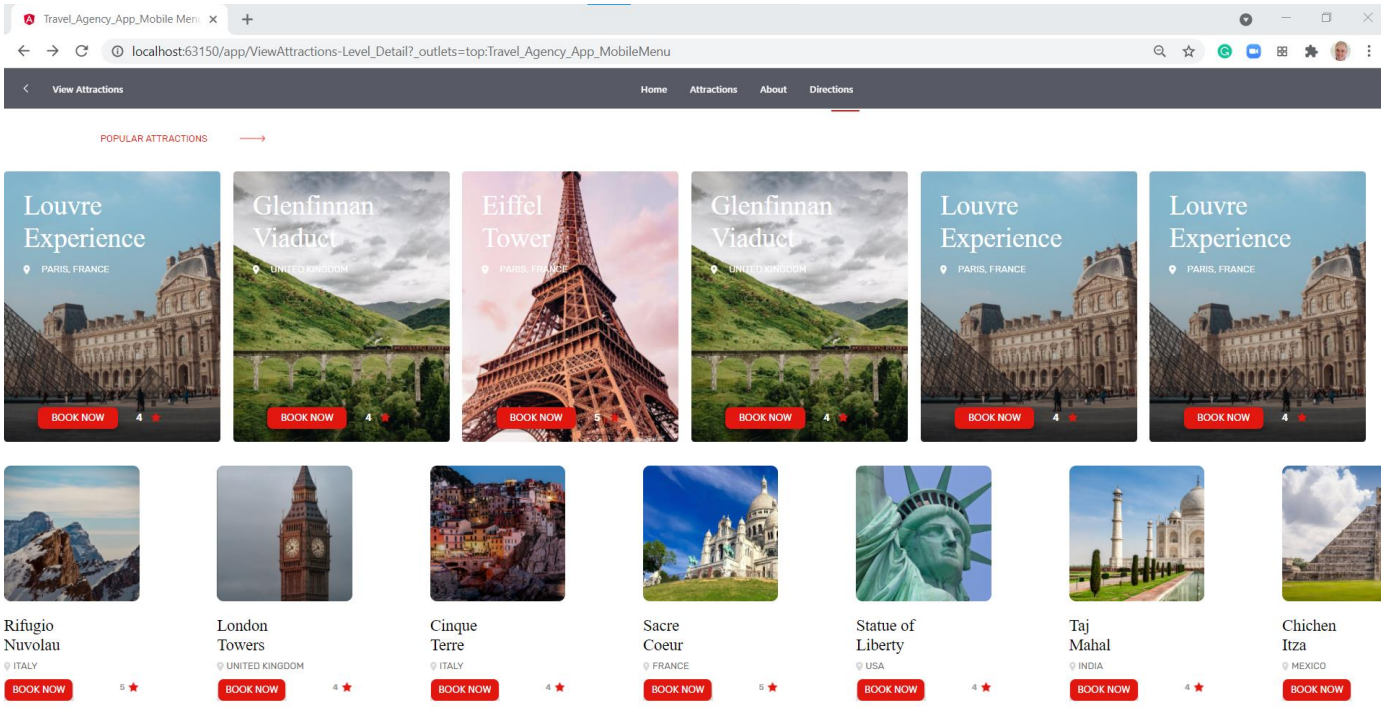


Abrimos GeneXus, pressionamos F5 e vemos que se abre o navegador web onde vemos a tela inicial da aplicação.

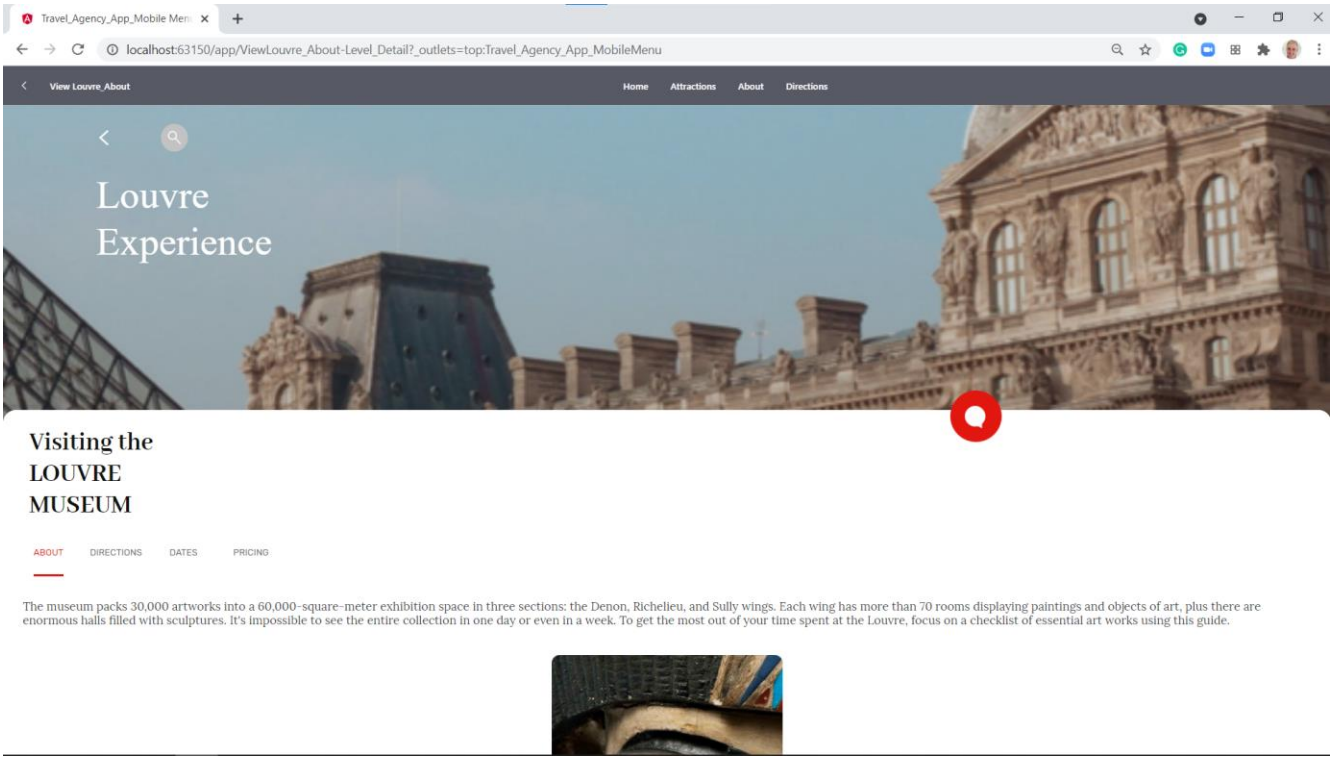
Na parte superior vemos o logotipo da agência e na parte cinza uma barra com botões de acesso rápido aos dados das atrações. No centro da página, vemos grandes botões que nos permitem acessar as diferentes partes da aplicação para visualizar informações sobre cruzeiros, voos, pacotes turísticos e atrações.



Se clicamos neste último botão, acessamos uma página onde podemos navegar horizontalmente entre as atrações mais populares. Para cada atração, vemos sua imagem, sua localização, sua qualificação e um botão para agendar uma visita.

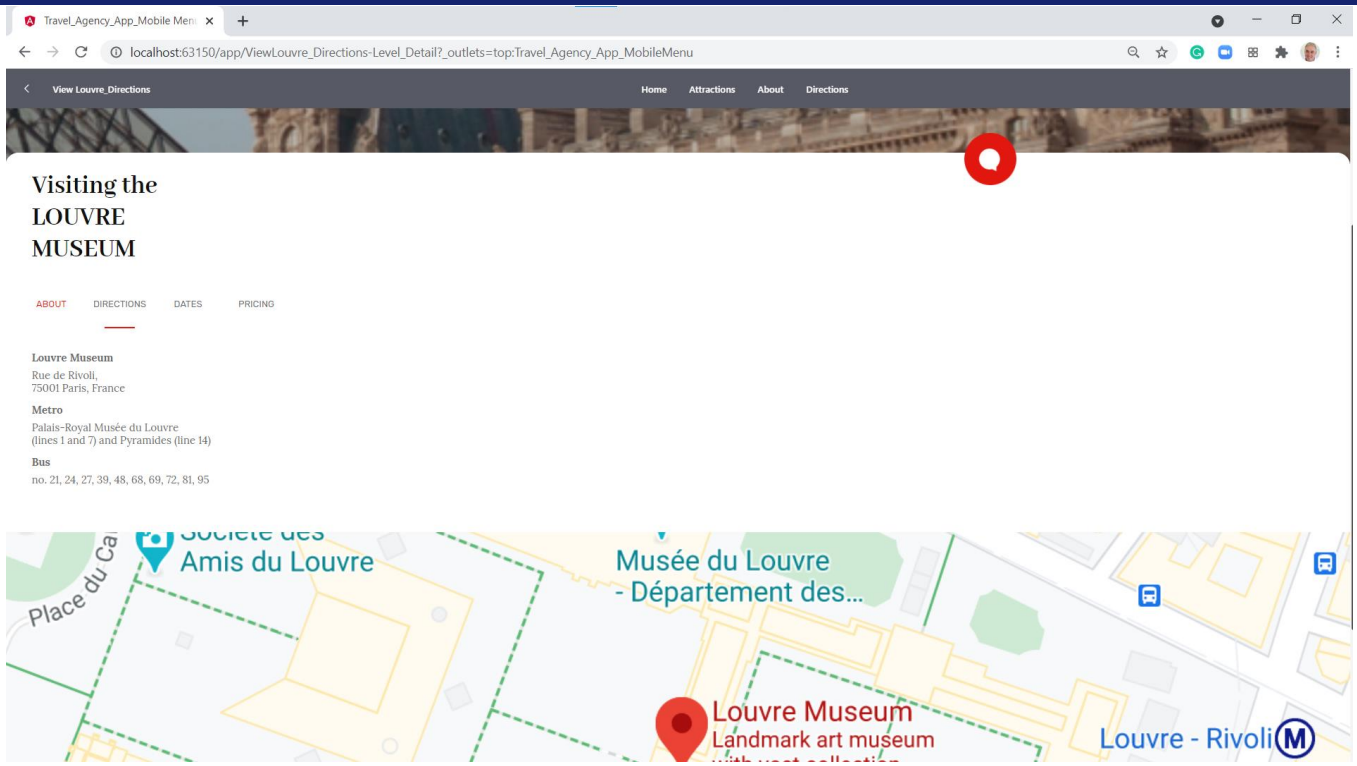


Mais abaixo na página, vemos outra lista com todas as atrações para visitar. Aqui também podemos nos mover horizontalmente e para cada atração vemos seu país, qualificação e botões para marcar uma visita.



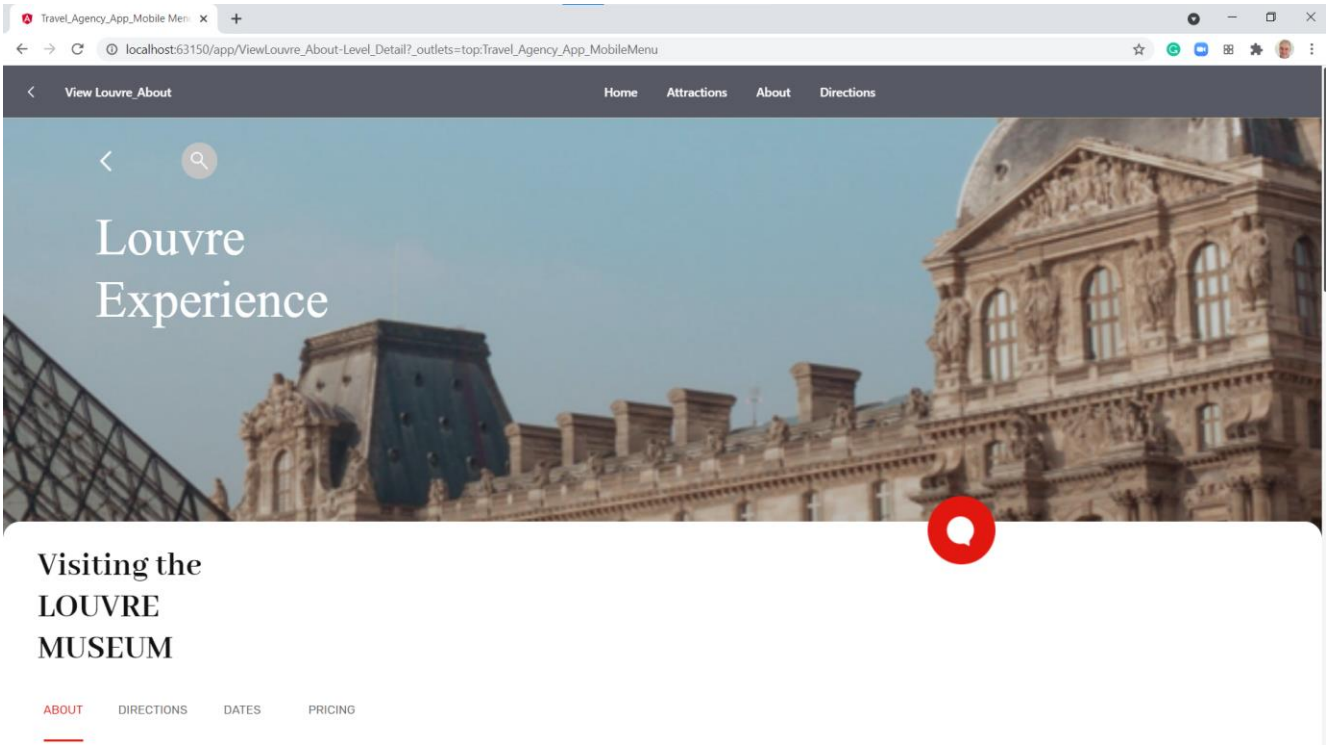
Se clicamos em uma atração popular, por exemplo, no Louvre, vemos que se abre uma página com o detalhe da atração, onde podemos ler uma breve descrição com imagens representativas.

Vemos também um submenu, onde além destes dados gerais (marcado com About) podemos acessar outros dados da atração como seus endereços, datas das visitas e seus preços.



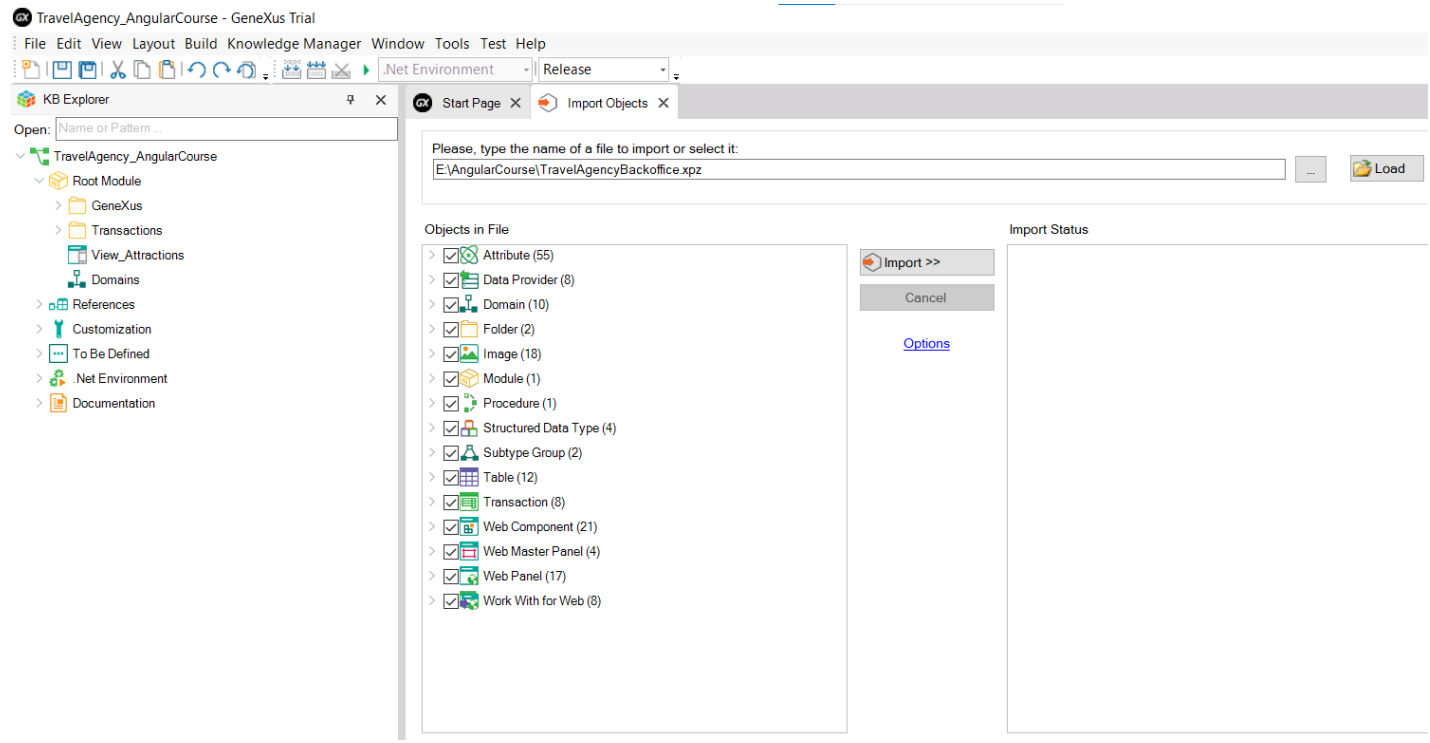
Se selecionamos endereços, vemos a rua onde está o museu do Louvre, sua caixa postal, cidade e país. Também vemos as linhas e paradas do metrô e as linhas de ônibus para acessar o museu.

Mais abaixo, vemos em um mapa a localização precisa da entrada.



Se pressionamos o botão da barra superior que diz Attractions, voltamos para a lista de atrações. E se pressionamos home, voltamos à tela inicial.

Nos vídeos a seguir veremos como completaremos estas funcionalidades, a princípio sem levar muito em conta questões da aparência como cores, formas de botões e estilo de fontes, já que nos concentraremos no desenvolvimento do que deve fazer a aplicação e de sua lógica, e uma vez feito isto, veremos como importar para nossa aplicação as definições de estética, realizadas por um profissional de design que nos ajudará a alcançar a aparência desejada.



Voltamos ao GeneXus e começamos a desenvolver a aplicação que vimos. A primeira coisa que fazemos é criar uma KB chamada TravelAgency_AngularCourse.

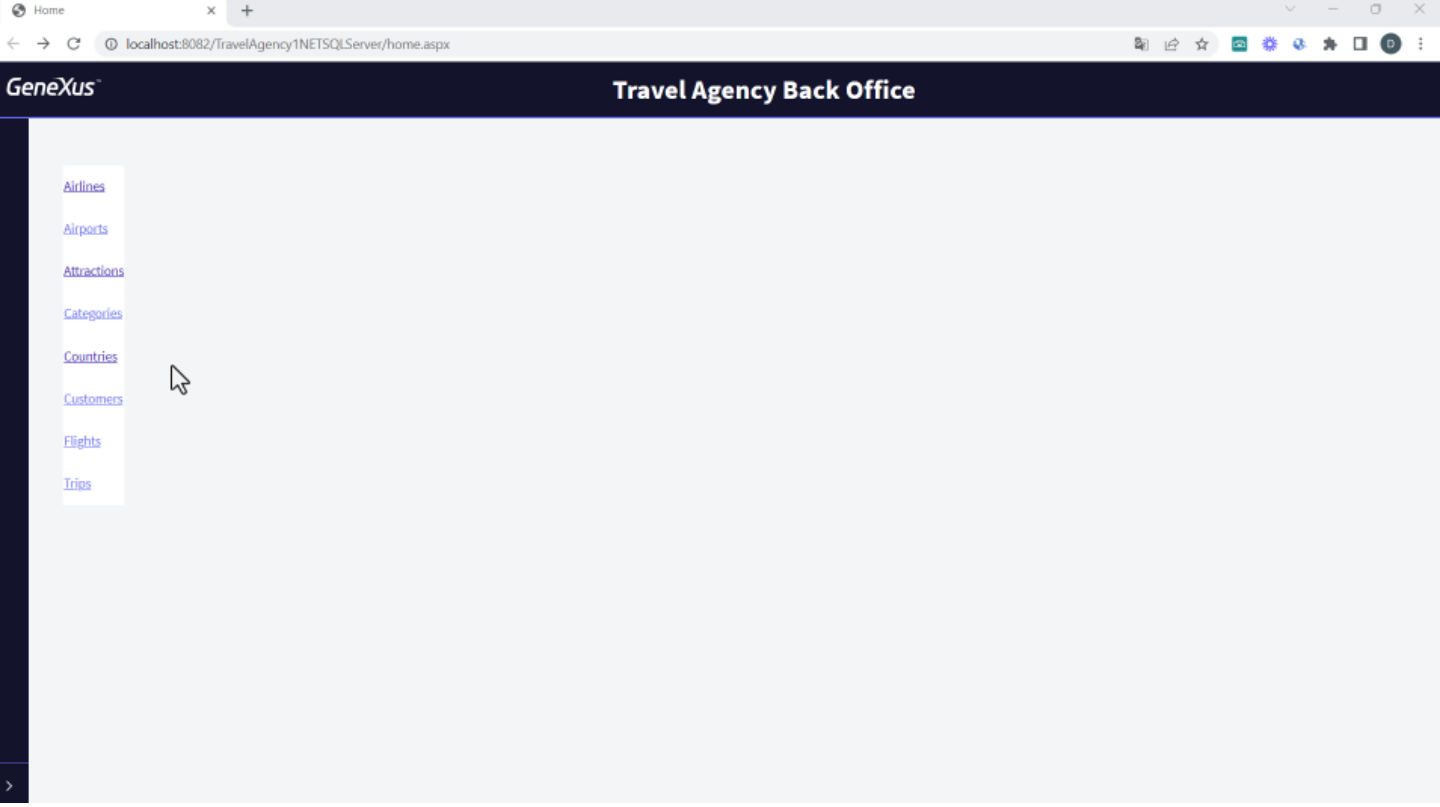
Agora importamos o modelo de dados já pronto, com todas as transações que utilizaremos, às quais já aplicamos o padrão WorkWith for Web, para que tenhamos disponível o backoffice da aplicação da agência de viagens.

Este backoffice web já está feito antes e nós o importamos para utilizá-lo como está, já que sua implementação não é o objetivo do curso. O que vamos desenvolver durante o curso é o front-end web voltado para customer-facing, em Angular.

Vamos para Knowledge Manager/Import e selecionamos o arquivo TravelAgencyBackoffice.xpz

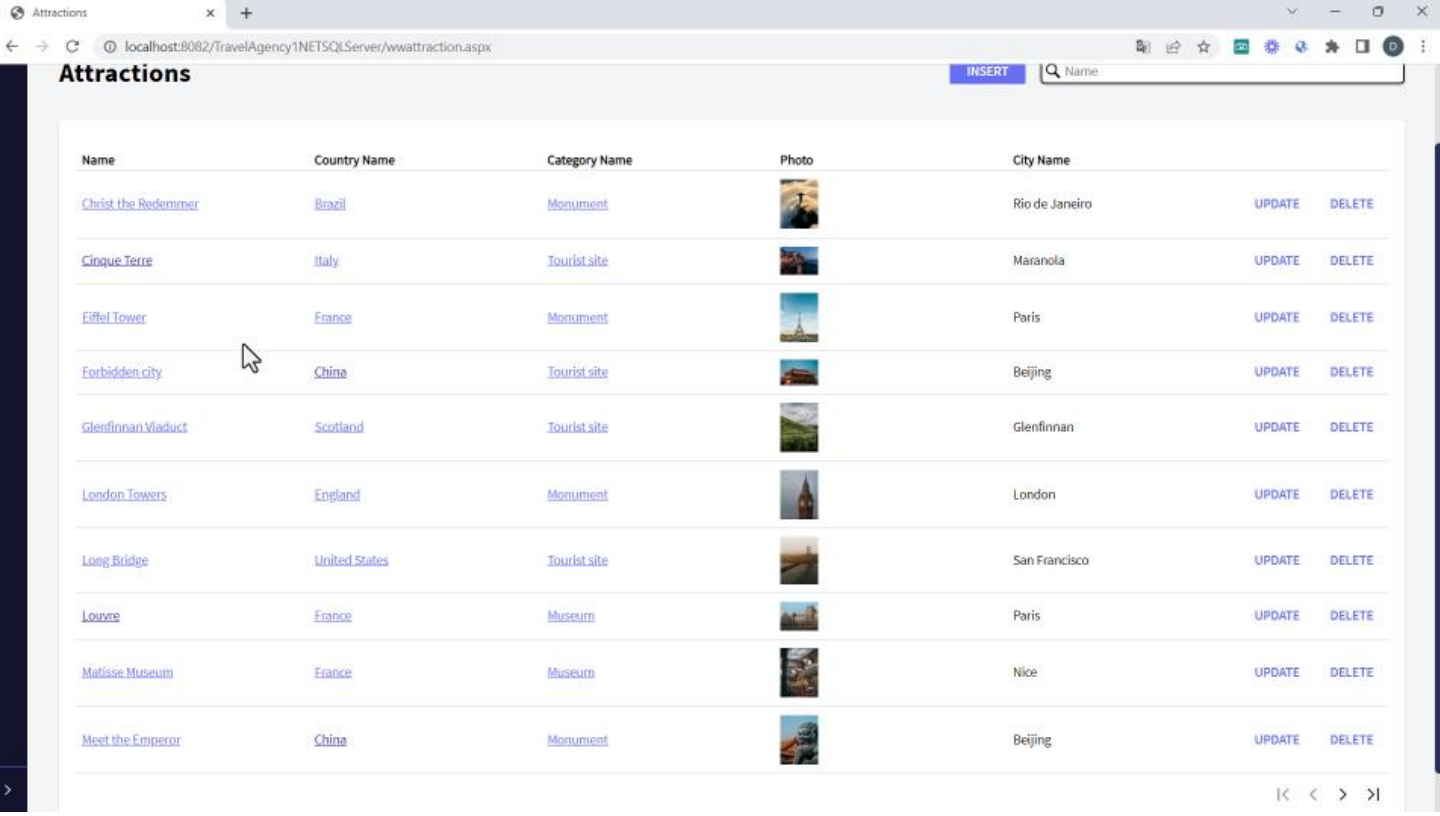
Este xpz contém também data providers de carga de dados, portanto, ao executarmos será criada a base de dados, com suas tabelas já com dados.

Primeiro selecionamos o web panel Home como startup object e agora pressionamos F5 para nos familiarizar com as entidades da agência de viagens enquanto executamos o backoffice da aplicação.



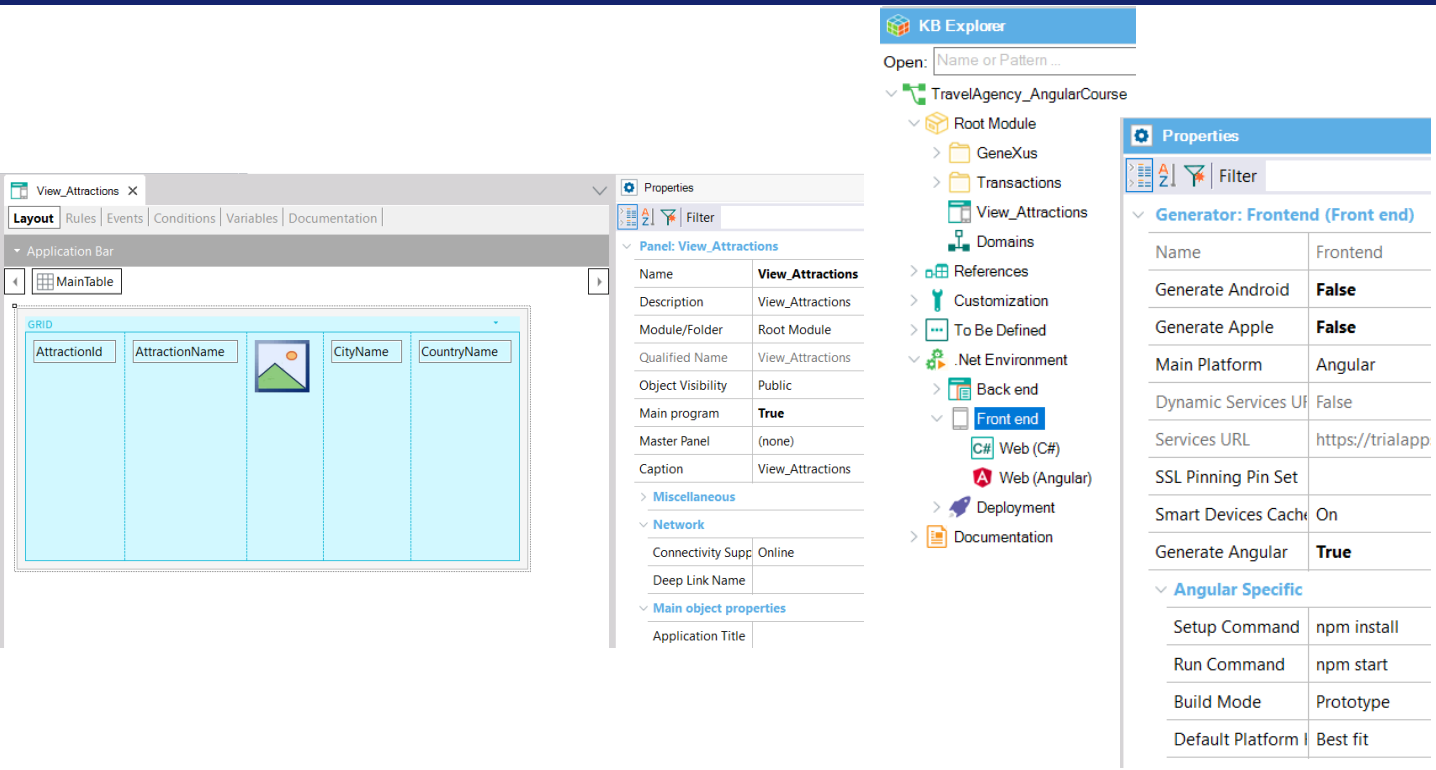
Neste web panel Home, vemos um link para cada entidade. Se formos aos países, vemos que estão carregados vários países, cada um com suas cidades. Se formos para as atrações turísticas, vemos que atração pertence a um país e a uma cidade e tem informações detalhadas da atração.

Se voltamos ao Home, podemos navegar pelos demais dados da agência de viagens. Se tem dúvidas sobre os objetos transação, o padrão work with ou os data providers, sugerimos rever estes conceitos no curso GeneXus Core.



Se formos para as atrações turísticas, vemos que atração pertence a um país e a uma cidade e tem informações detalhadas da atração.

Se voltamos ao Home, podemos navegar pelos demais dados da agência de viagens. Se tem dúvidas sobre os objetos transação, o padrão work with ou os data providers, sugerimos rever estes conceitos no curso GeneXus Core.



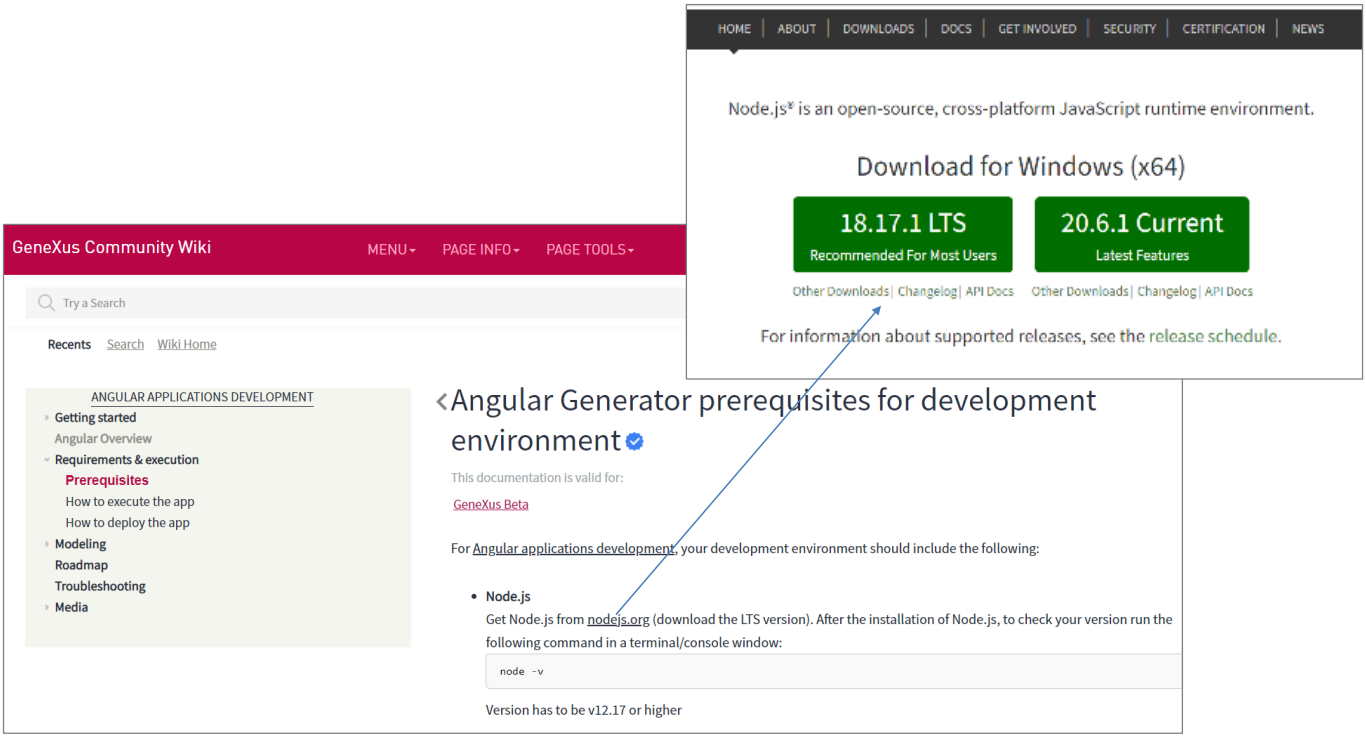
Voltamos ao GeneXus e começamos com a implementação da primeira tela da aplicação.

A agência de viagens nos solicita construir um panel que mostre uma lista horizontal das atrações disponíveis e quando selecionamos uma, ver os detalhes da atração.

Para isso, criamos um objeto Panel, que chamamos de View_Attractions.

Arrastamos para o form um grid e selecionamos os atributos AttractionId, AttractionName, AttractionPhoto, CityName e CountryName.
Vamos para as propriedades do panel e o definimos como objeto main.

Agora no Knowledgebase Navigator clicamos duas vezes no environment .NET, selecionamos Front end e vamos para suas propriedades. Definimos Generate Angular em True e, em seguida, Generate Android e Generate Apple em False. Observamos que a propriedade Main Platform ficou com o valor Angular.



Antes de gerar nossa aplicação em Angular, devemos instalar algum software. Podemos ver isto se formos à wiki e procurarmos: “Angular requirements”.

Vemos que temos que instalar o Node.js a partir da página nodejs.org. Aqui, baixamos e instalamos a versão LTS. Já fiz isso em minha máquina.

For Angular Applications Development, your development environment should include the following:

- **Node.js**

  Get Node.js from nodejs.org (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

  ```
  node -v
  ```

- **npm package manager**

  For the environment setup, npm packages will be installed (automatically) using the npm client command line interface, so you must have an npm package manager. Node.js includes it by default. So, after installing Node.js, check that you have the npm client installed, by running the following command in a terminal/console window (npm version should be 7 or higher):

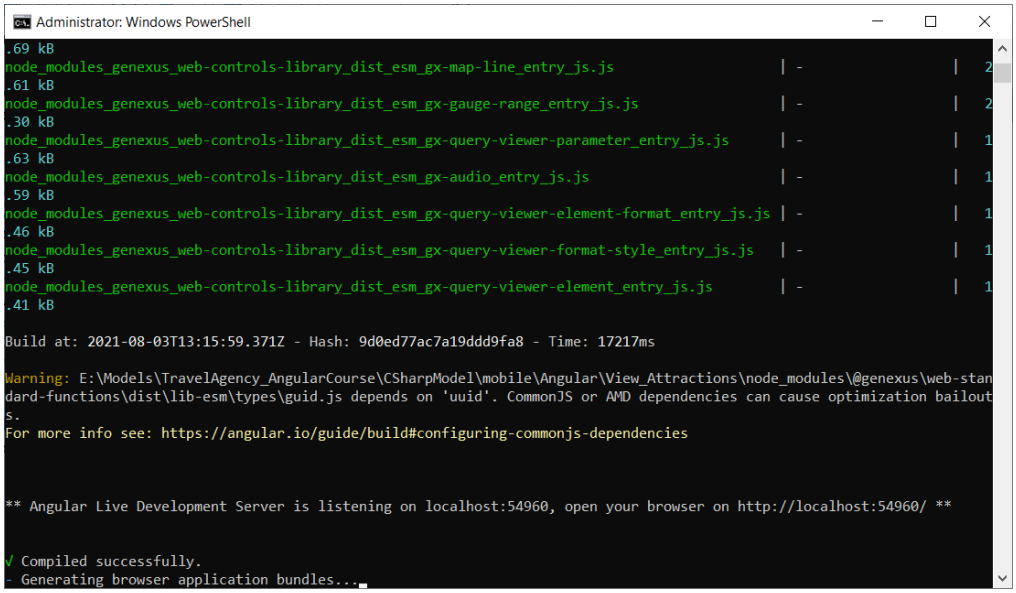
  ```
  npm -v
  ```

Uma vez instalada, para verificar a versão, abrimos uma janela de comando e escrevemos: **node –v**

Em seguida, temos que instalar um cliente gerenciador de pacotes npm, mas como já temos instalado o Node.js, ele já o inclui.

Para verificar se foi corretamente instalado, escrevemos “npm –v” e obtemos a versão.

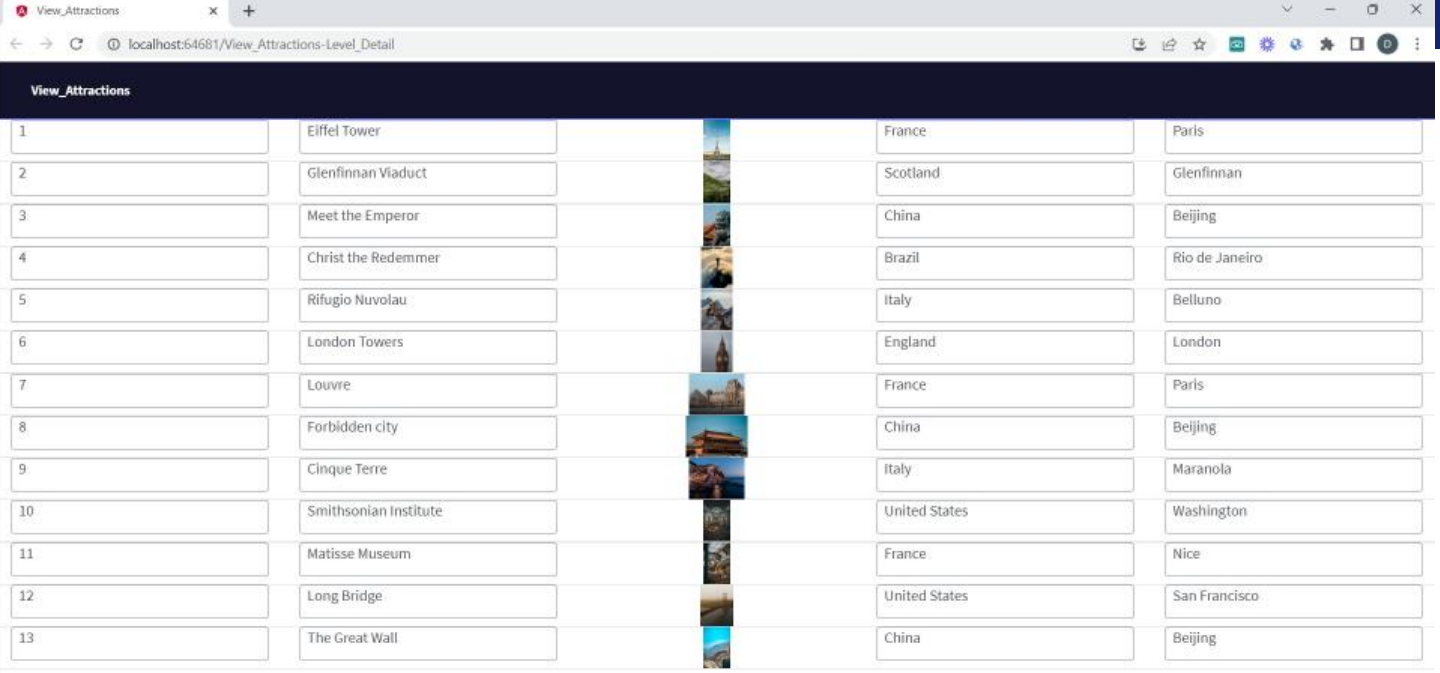
Agora sim, temos tudo pronto para gerar e executar nossa aplicação em Angular.



Voltamos ao GeneXus, vamos ao panel View_Attractions, clicamos com o botão direito e selecionamos Run.

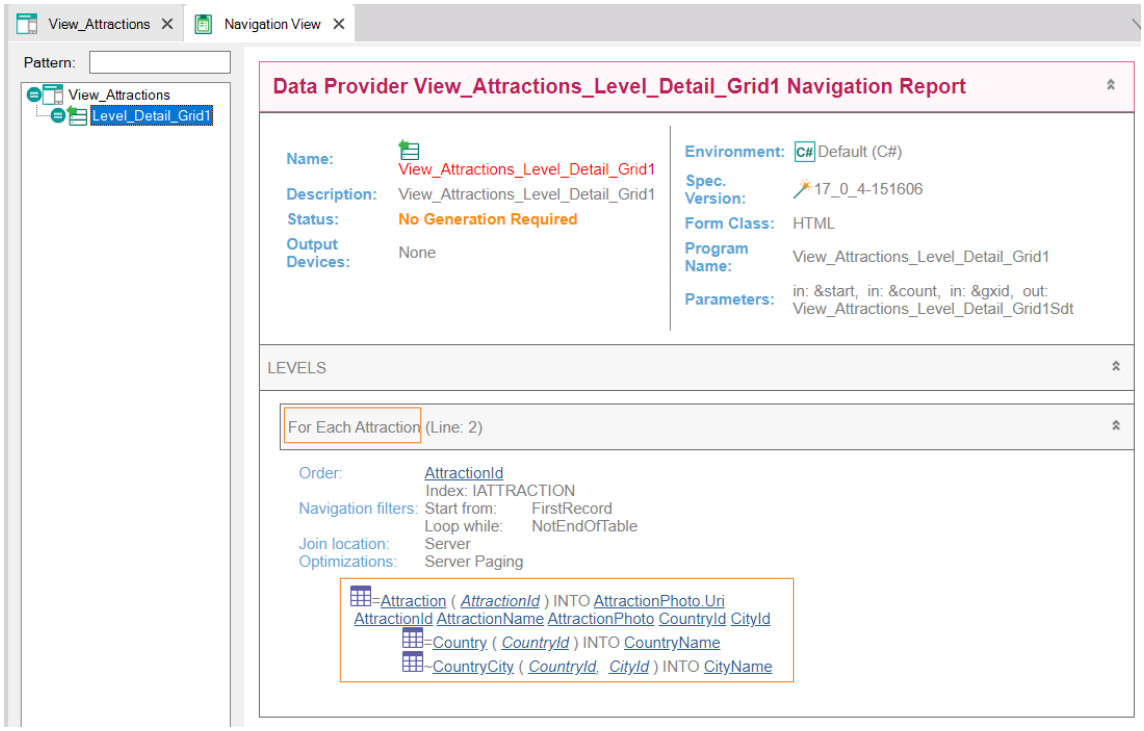
Vemos que se abre uma janela de comandos e que são executadas várias linhas escritas pelo gerador Angular.

Ao terminar, é aberto automaticamente o navegador web e vemos a lista de atrações turísticas.



Já temos nossa primeira aplicação Angular funcionando.

Se observamos a URL, vemos que foi executada em localhost, dois pontos e uma porta atribuída dinamicamente. Isto corresponde ao servidor web instanciado para executar a aplicação Angular. A aplicação executada é View_Attractions-Level_Detail, que corresponde ao grid de atrações que incluímos no panel.



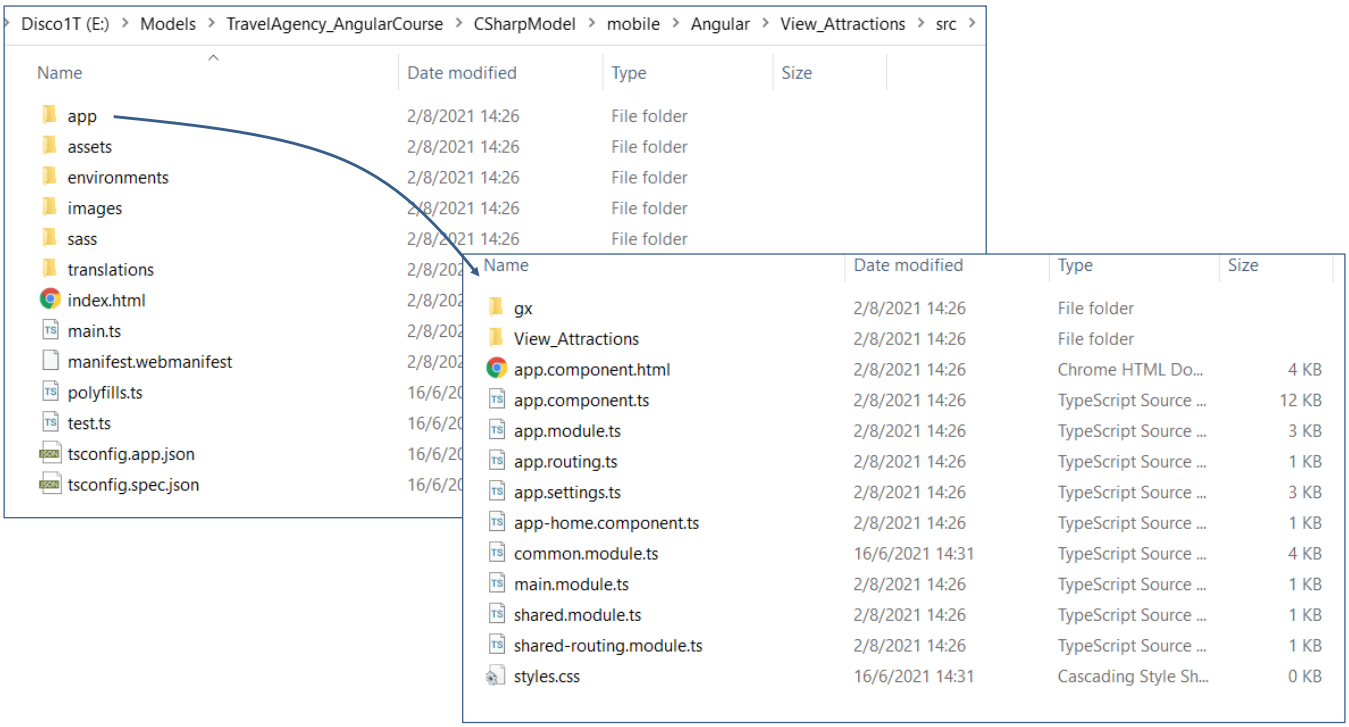
Se voltamos ao GeneXus e vemos a lista de navegação, vemos que o panel View_Attractions contém um nó chamado Level_Detail_Grid1 e que tem o símbolo de um data provider.

Se clicamos sobre ele, vemos que é a lista de navegação do data provider View_Attractions_Level_Detail_Grid1 que é o responsável por acessar a base de dados e recuperar as informações das atrações turísticas.

Se vemos onde diz Levels, encontramos um For Each Attraction já que como nosso grid tinha atributos, GeneXus encontrou uma tabela base e criou um for each implícito para acessar os dados da base de dados.

Vemos que efetivamente foi navegada a tabela Attraction para trazer os dados da atração e, em seguida, foi acessada a tabela Country para recuperar o nome do país e a tabela CountryCity para obter o nome da cidade.

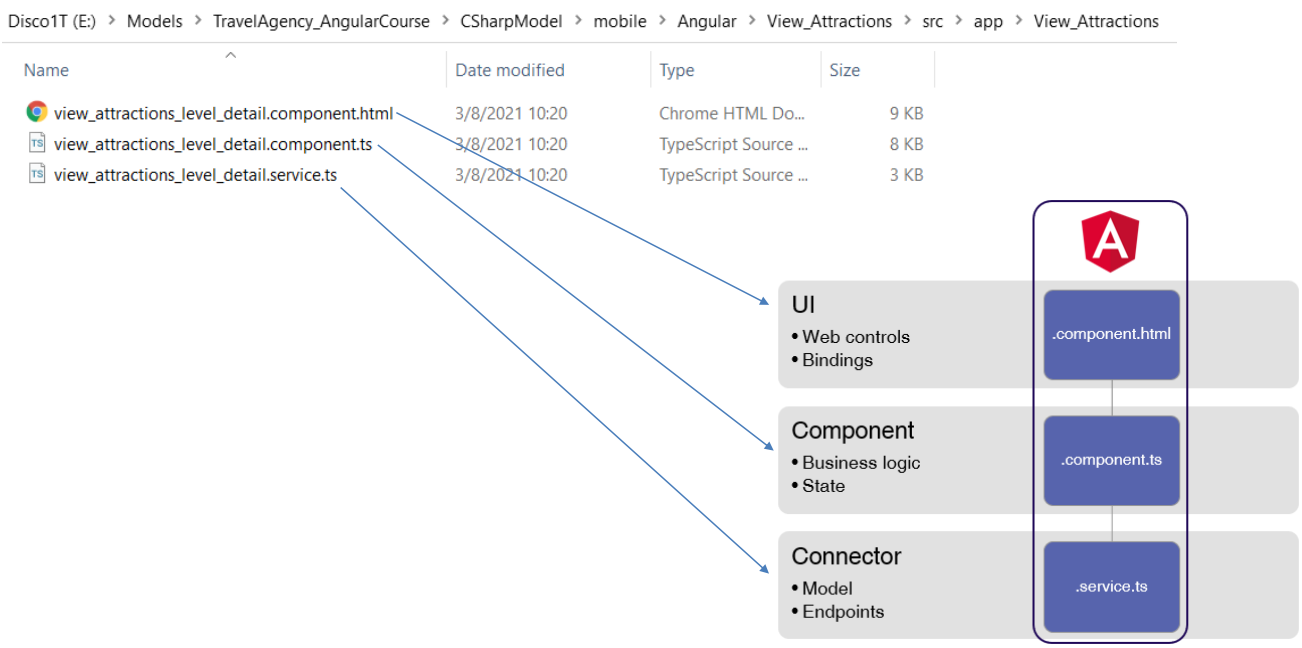
Mais adiante veremos o que leva em consideração GeneXus para determinar a tabela base do grid e como se carrega a informação através do data provider cuja lista de navegação estamos vendo.



Vamos analisar agora o que gera GeneXus, quando é executado o gerador Angular.

Se vamos para Tools/Explore Target Environment e depois vamos para mobile/Angular, encontramos uma pasta View_Attractions. Isto ocorre porque é o objeto main que executamos. Se agora selecionamos src/app, vemos as partes que compõem a aplicação.

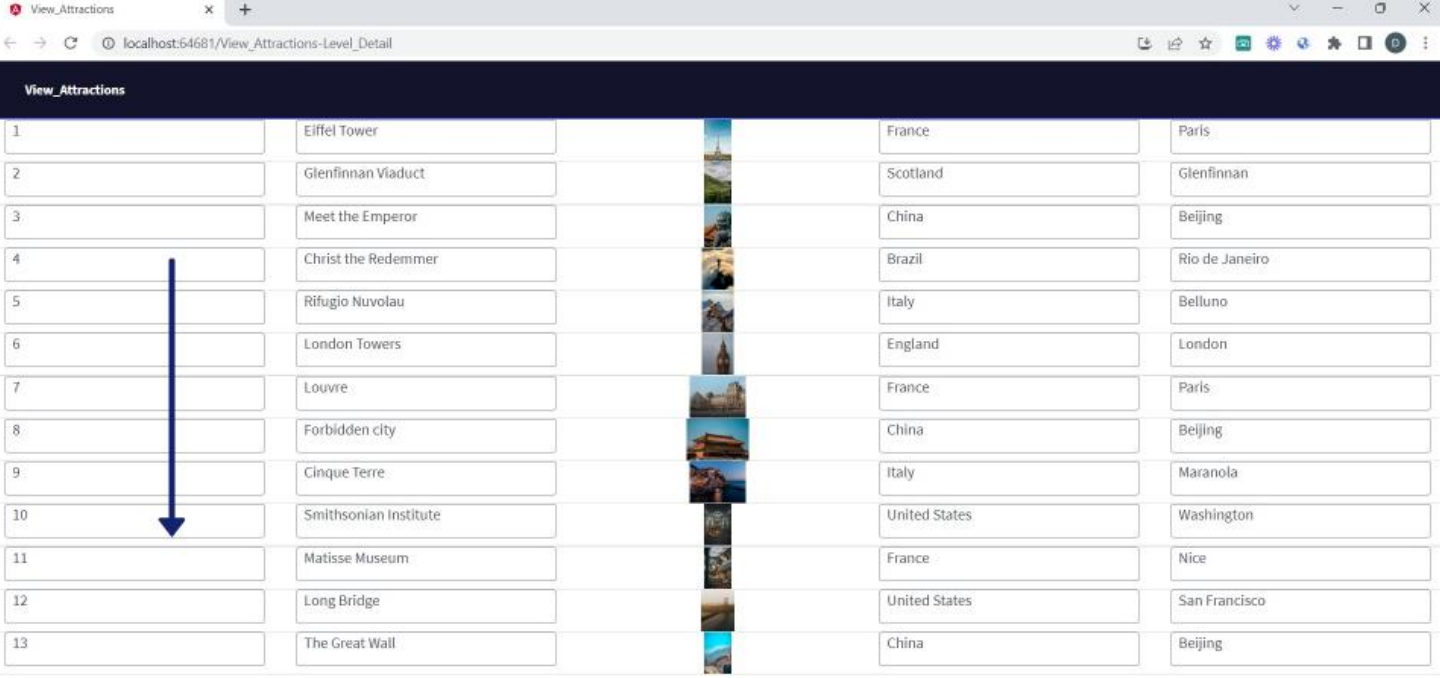
Se clicamos em View_Attractions, vemos as partes que incluem o componente Angular, que são 3 arquivos.



O arquivo que termina em **component.html** contém a definição da interface de usuário da aplicação e é um template de Angular que representa o layout do panel. Aqui também estão incluídos elementos que Angular interpreta para associar os controles em tela com os dados da base de dados.

Então temos o arquivo que termina em **component.ts**, que é um arquivo TypeScript onde está definida a lógica do panel que implementamos. É aqui onde estão definidos os eventos e o estado dos controles da tela.

Por último, temos o arquivo que termina em **service.ts** que é responsável por interagir com os serviços do servidor web. Aqui ocorrem as comunicações com os serviços REST, são obtidos os dados da base de dados e são geradas as estruturas de dados que o panel vai utilizar.



Se voltamos para a tela de atrações, vemos um grid que mostra os dados das atrações para baixo, que é o modo padrão.

No desenvolvimento do curso veremos como alterar esta visualização para poder paginar para a direita, bem como outros controles de tela que nos permitirão construir a aplicação com uma interface de usuário mais atrativa.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications