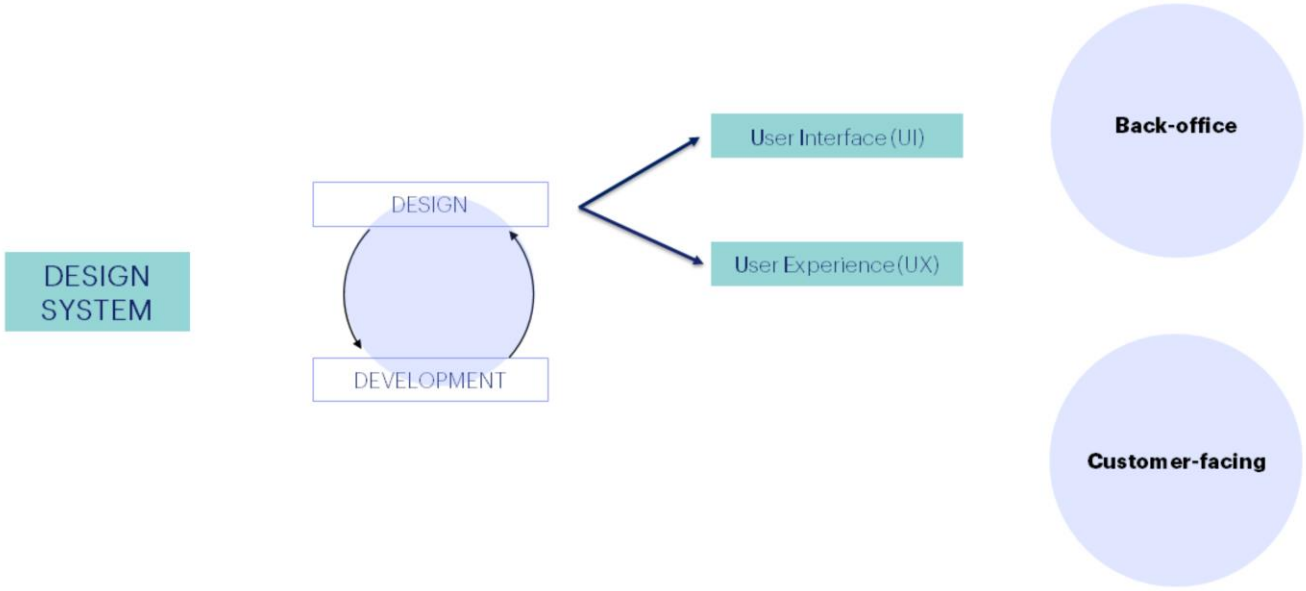
Telas web com foco em Customer-facing

Design System

GeneXus™

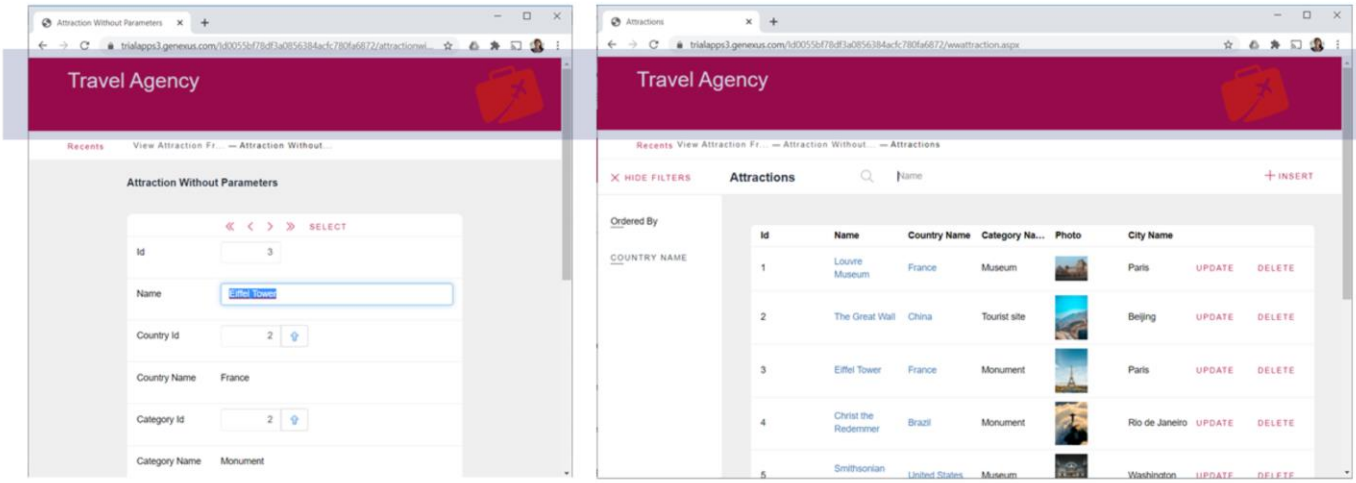
Até agora não nos preocupamos com o desenho de nossos painéis, mas nos concentramos em seu funcionamento. Vejamos agora como incorporar desenho à nossa aplicação.



Nos vídeos anteriores vimos que tanto as aplicações de Back-office quanto as Customer-Facing têm requisitos de desenho e usabilidade que levaram à introdução do conceito de Design System.

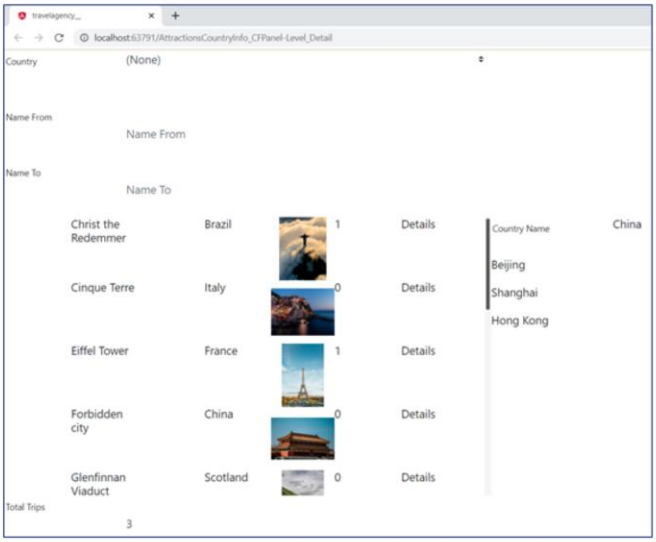
O Design System envolve o conjunto de Princípios, Padrões e Práticas que proporcionam à nossa aplicação coerência, uniformidade e robustez. Isto determina que o ciclo de desenvolvimento envolva tanto desenvolvedores, quanto designers.

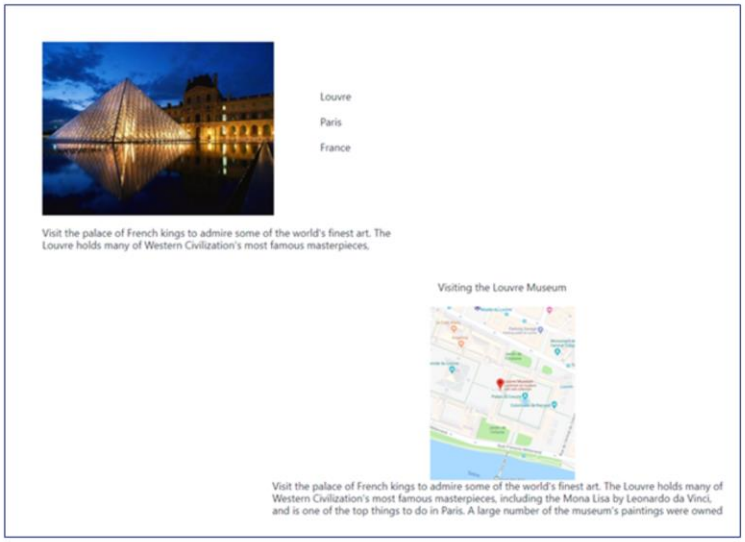
Design System predeterminado em uma Web app



Vimos também que tanto as transações como os webpanels que integravam o back-office, contavam com um Design System predeterminado.

Design System predeterminado em customer-facing



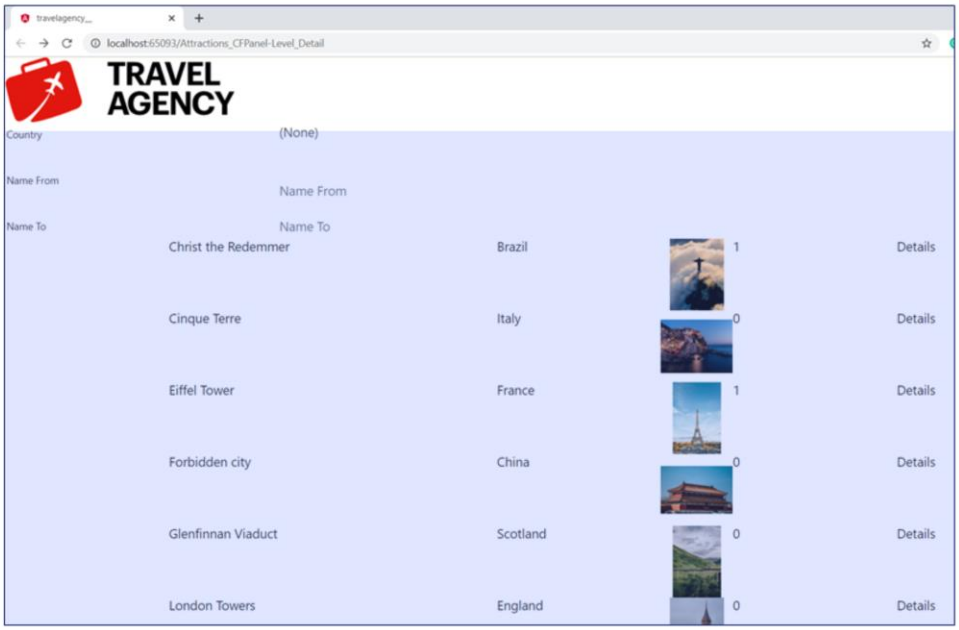


Como na aplicação de back-office, nos objetos que usamos na parte customer-facing, há também um desenho predefinido incorporado por GeneXus.

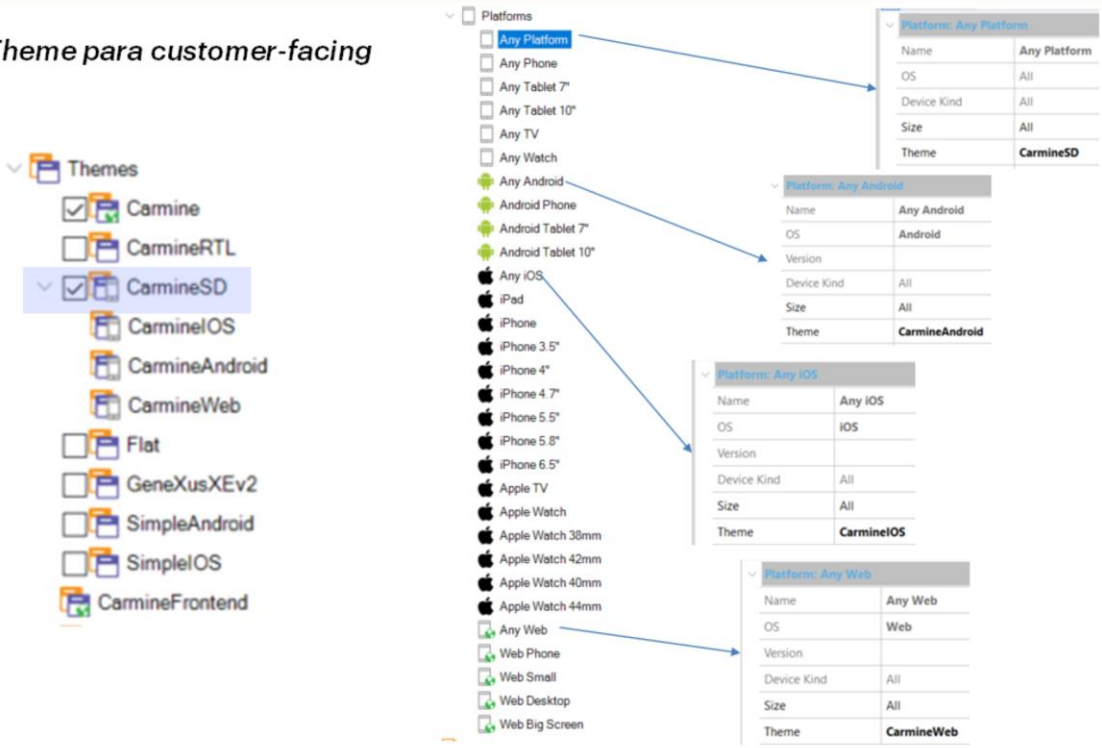
Ao contrário dos objetos transação e webpanels que tinham atribuído por padrão uma master page, nos objetos do front-end customer-facing, não há um objeto Master Panel atribuído por padrão que forneça um contentor onde se execute a aplicação
e incorpore componentes básicos de desenho na execução.

No entanto, podemos observar que os controles seguem um desenho e têm uma coerência de aspecto, de forma predeterminada. Também vimos que é possível atribuir um objeto Master Panel aos painéis da aplicação.

Design System predeterminado em customer-facing



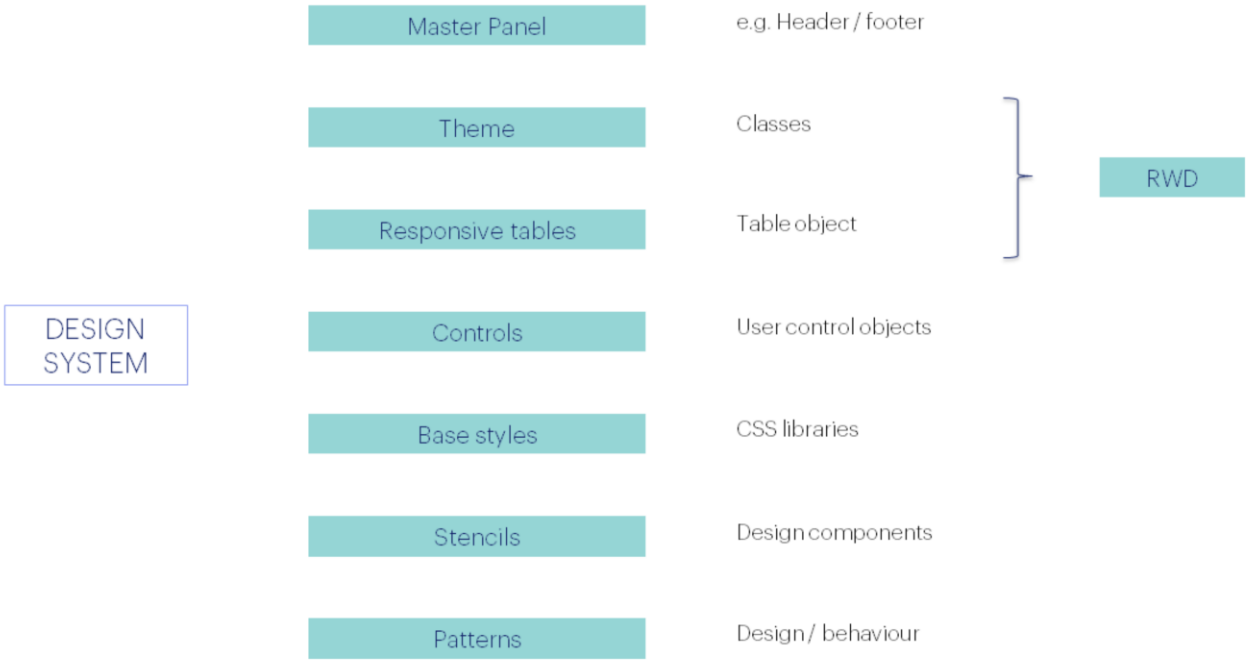
Objeto Theme para customer-facing



A aparência dos controles na tela são tomadas a partir das definições do objeto Theme de nome CarmineSD.

Depois, dependendo da plataforma escolhida, será utilizado um dos subtemas, por exemplo, quando geramos em Android, utiliza-se CarmineAndroid, para Apple utiliza-se CarmineIOS e quando geramos Angular, assumem-se as definições do tema CarmineWeb.

Isto podemos verificar se no KB Explorer, abrimos o nó Platforms e para cada plataforma, vemos sua propriedade Theme. Por exemplo, para Any Platform vemos que está atribuído o tema CarmineSD, para Any Android o tema CarmineAndroid, para Any iOS o tema CarmineIOS e para Any Web o tema CarmineWeb.



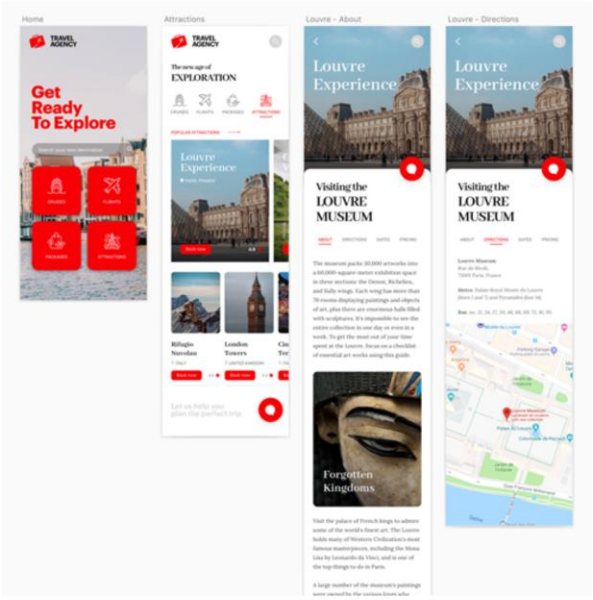
Em relação aos componentes que vimos que integram um Design System, em uma aplicação customer-facing, também contamos com um master panel, objetos theme e classes. Não há um controle dedicado a tabelas responsivas, já que o controle Table permite a adaptação automática do conteúdo quando se gera uma aplicação web com Angular.

Diferente do padrão Work With para web que gera objetos webpanels, o padrão Work With para objetos customer-facing não gera objetos panels, mas gera um objeto específico denominado WorkWithDevices.

O resto dos componentes do Design System estão representados, além disso a implementação de cada um varia com a plataforma (web ou nativa) e com o gerador utilizado (Angular, Android ou Apple).

Como quando vimos a personalização do desenho na aplicação de back-office, nas aplicações customer-facing é possível trabalhar com os diferentes elementos do Design System (classes, temas, controles, etc.) para alcançar um desenho que permita a melhor experiência de usuário.

Importação de um desenho a partir do Sketch

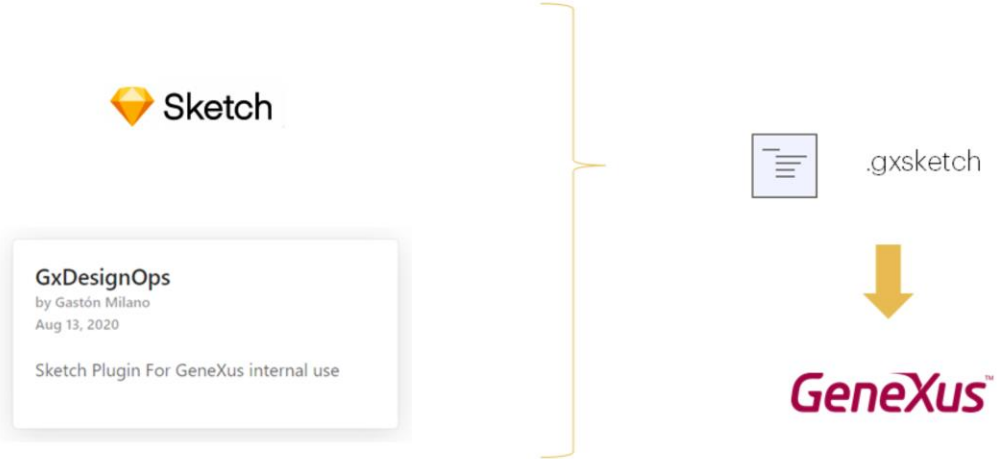


No entanto, o melhor caminho e o mais simples, é que um designer defina isso por nós e importemos o desenho em nossa KB.

Queremos criar uma página inicial com botões para diferentes partes da aplicação, mas apenas nos interessa ver os dados das atrações turísticas. Ao pressionar o botão das atrações queremos ver uma lista das atrações disponíveis e, depois, clicar sobre uma atração para poder ver seus detalhes e sua localização em um mapa.

Pedimos ao designer que nos enviasse um desenho em princípio para um dispositivo móvel e veremos que depois também poderemos gerar a aplicação em Angular usando os mesmos objetos criados na importação.

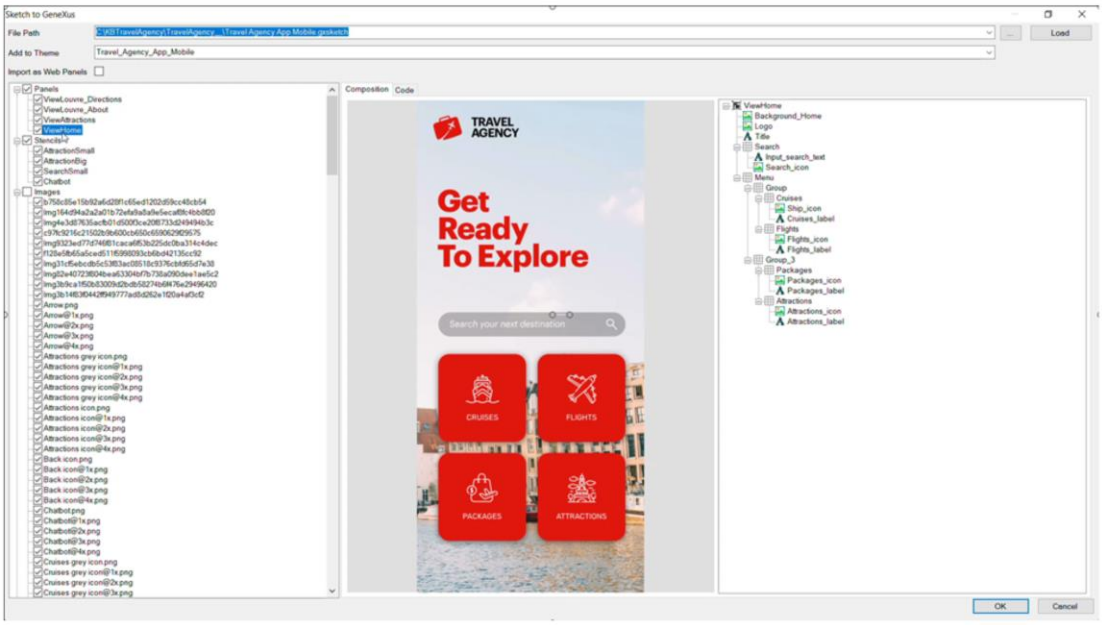
Importação a partir do Sketch



Para cumprir com o que pedimos, o designer criará um desenho usando a ferramenta Sketch e, em seguida, utilizará um plugin GeneXus instalado na mesma, que permite criar um arquivo de extensão gxsketch, que é o que nos envia.

O que fazemos a seguir, é importá-lo em nossa KB e se integrarão todos os componentes de desenho e inclusive serão criados os objetos GeneXus correspondentes para conter as definições.

Importação do arquivo .sketch



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

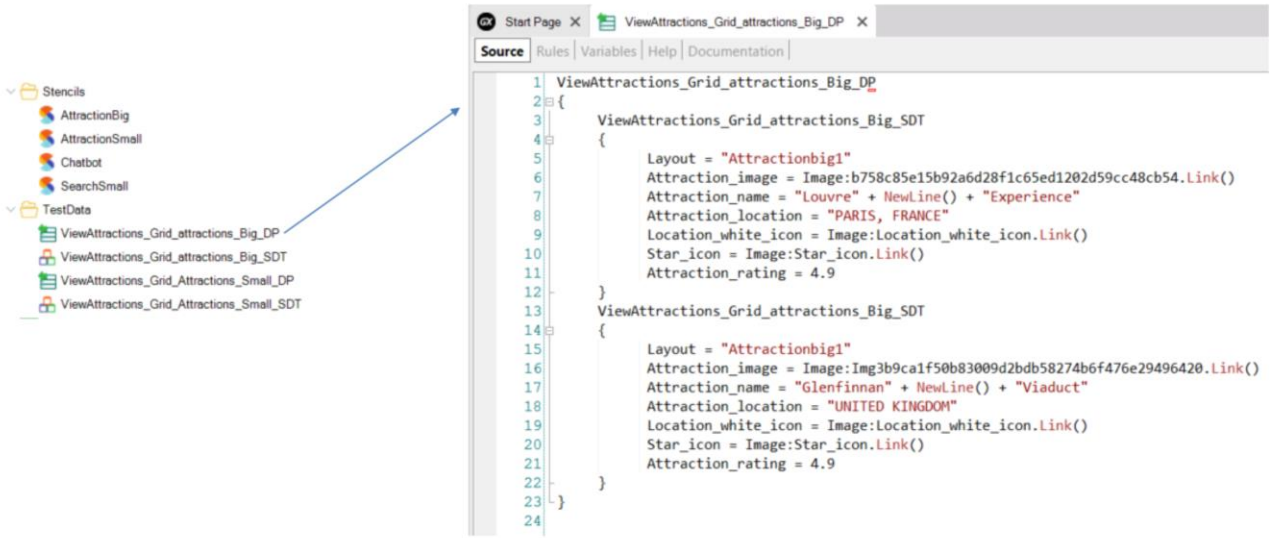
Deixamos o check box Import as Web Panels desmarcado, porque queremos que nos crie objetos panel e não webpanels.

Se clicamos nos panels que a importação criar, vemos a sua visualização prévia e confirmamos que correspondem ao desenho que validamos. À direita nos informa os controles que conterá.

Vemos que também importará as imagens para que possamos executar o panel com dados fixos e exibir atrações carregadas, que depois, deveremos substituir pelas armazenadas na base de dados da nossa aplicação.

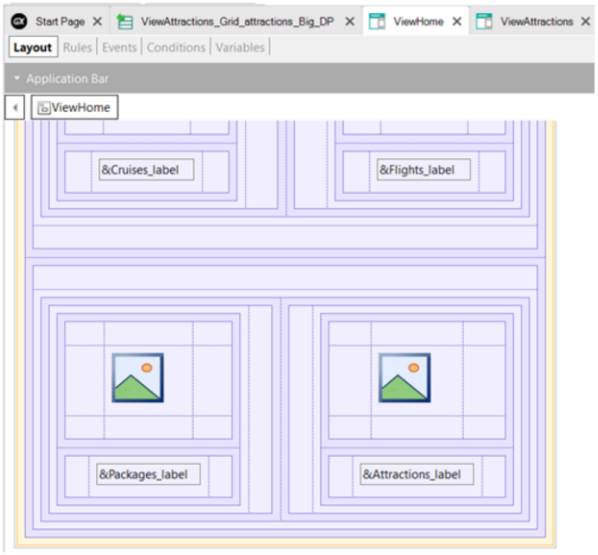
E mais abaixo vemos as fontes que foram utilizadas no desenho. Estamos de acordo, então pressionamos OK.

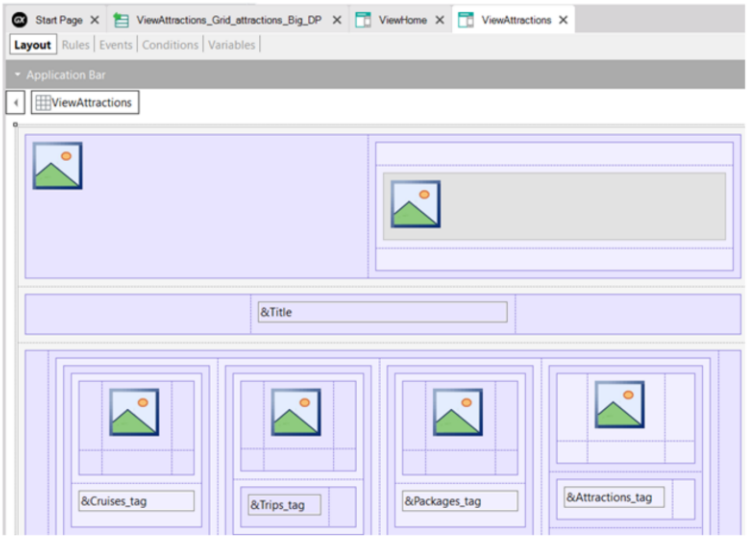
Objetos criados na importação



Na janela de Output é mostrado o progresso da importação e o resultado sem erros.
No KB Explorer, vemos que foram criados dois folders: Stencils com alguns stencils utilizados para encapsular desenho na aplicação e Test Data, que se o abrirmos vemos que contém data providers e sdt para carregar dados fixos.
Se agora abrimos o primeiro data provider, vemos o código de carga das imagens.

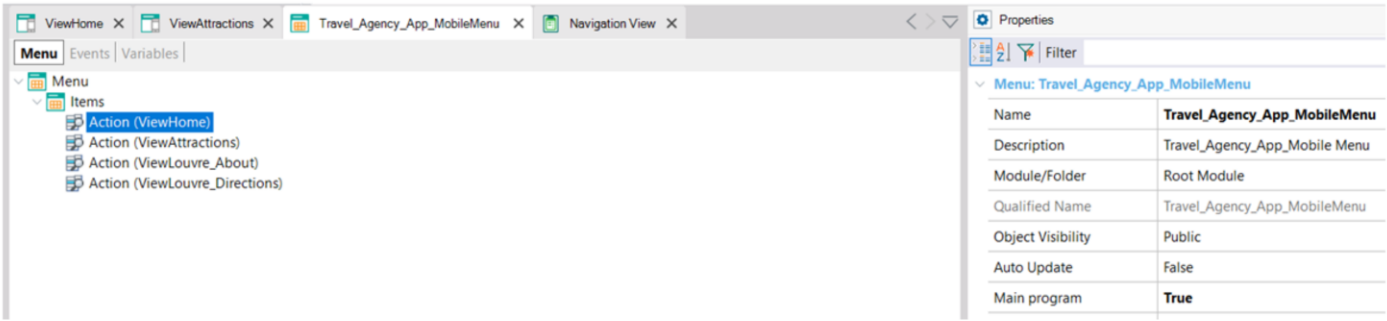
Objetos panels criados automaticamente na importação





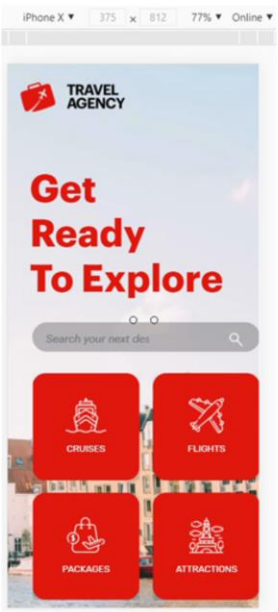
Vemos que foram criados automaticamente os objetos panels que tínhamos visto na tela do wizard de importação, como o da página inicial (ViewHome) e também o ViewAttractions.
Se vemos o form do ViewHome, observamos que já estão todos os componentes criados para o conteúdo visual, o mesmo para o ViewAttractions.

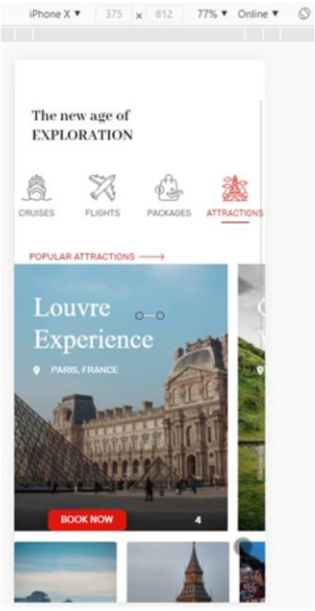
Objeto main de nossa aplicação



Aqui podemos ver que também foi criado um objeto menu que será o objeto main que vamos executar.
Este objeto será visto somente na geração para plataforma nativa, no caso de Angular, o objeto que será visto primeiro é o ViewHome, que é a primeira opção do menu.
Para executar, colocamos este objeto como Startup Object e damos F5.

A aplicação em execução com dados fixos



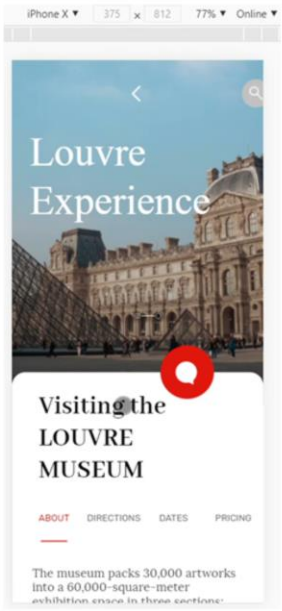


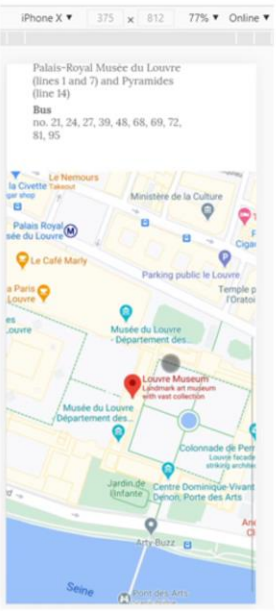
Como nosso desenho foi feito inicialmente para a plataforma mobile, escolhemos um tamanho de tela para iPhone X. Depois, teremos que fazer as alterações necessárias para poder vê-las em uma tela para desktop.

Vemos que é executada a tela inicial com botões para diferentes partes de nossa aplicação. Clicamos em Attractions e vemos a lista de atrações populares e, mais abaixo, outras atrações para visitar.

Recordemos que estamos vendo os dados fixos que carregaram os data providers da importação.

A aplicação em execução com dados fixos (cont.)

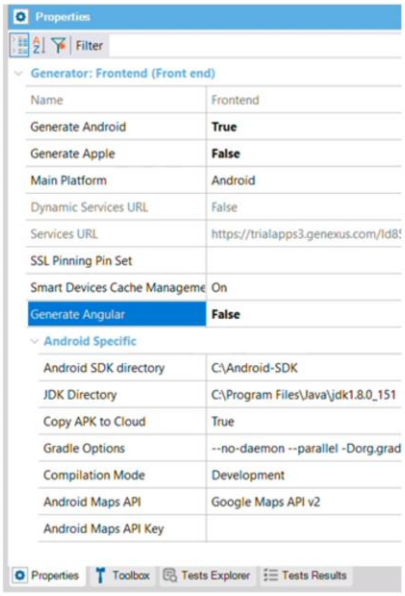




Se clicamos no Louvre, vemos que nos aparece uma página com informações detalhadas, tudo com um desenho elegante e esteticamente agradável.

Se agora clicamos em Directions, abre-se o panel com os dados do endereço do Louvre e um mapa com sua localização.

Geração e execução da aplicação customer-facing em Android



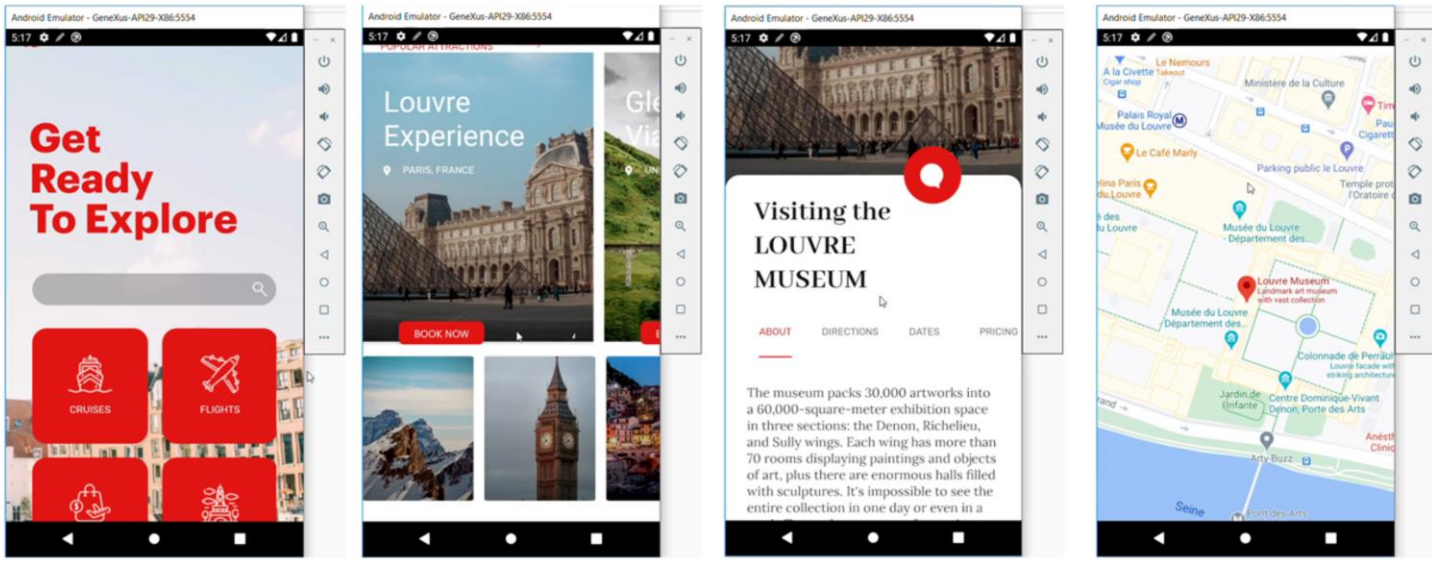
Agora geraremos a aplicação em Android para ver como fica com o novo desenho em uma plataforma mobile.

Para isso, vamos às propriedades do Front end e escolhemos Generate Android em True.

Como agora queremos ver apenas a aplicação mobile, vamos colocar a propriedade Generate Angular em False.

Como já tínhamos colocado o objeto menu como Startup Object, pressionamos F5.

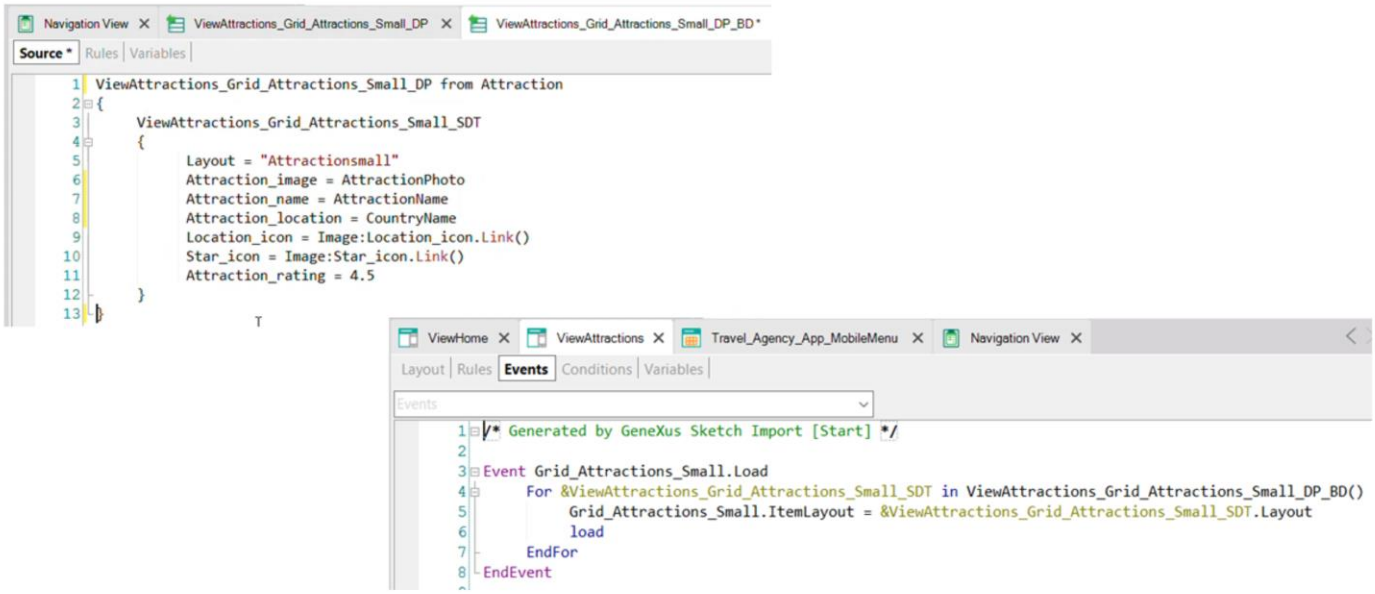
Geração e execução da aplicação customer-facing em Android



Vemos que se abre o emulador e a aplicação se executa em Android, com a visualização estética que esperávamos, conforme o Design System definido por nossos designers.

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

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



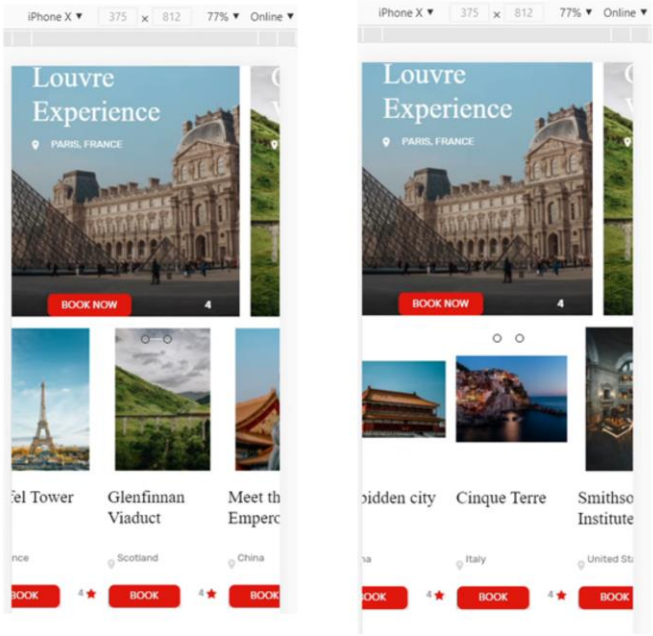
Até agora estivemos vendo dados fixos de teste, que os designers adicionaram para que possamos ver o comportamento da aplicação com o novo desenho. Então vamos fazer algumas mudanças, para que possamos ver os dados reais de nossa base de dados.

Se vamos ao panel ViewAttractions, vemos que o grid exibe as atrações que estão carregadas em uma variável tipo SDT de nome ViewAttractions_Grid_Attractions_Small_SDT. Se vamos aos eventos do panel, vemos que esta é carregada com um data provider. O abrimos, fazemos Save As e salvamos o novo data provider com o nome ViewAttractions_Grid_Attractions_Small_ 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, e ao Attraction_location atribuímos o atributo CountryName. Apagamos tudo o que não nos serve. Agora voltamos ao panel ViewAttractions e trocamos o data provider que carrega as atrações pelo novo que construímos.

Salvamos e vamos novamente habilitar a geração em Angular. Agora, clicamos com o botão direito em nosso objeto main e escolhemos Run.

Execução da aplicação com dados reais

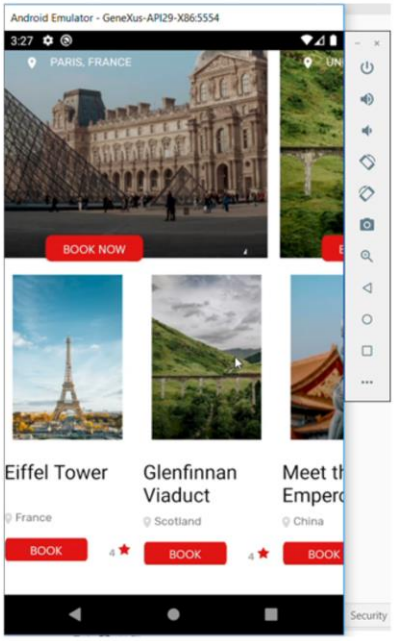


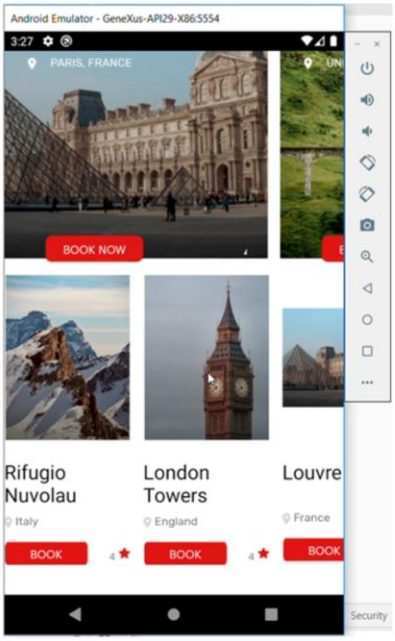
Se vamos às atrações, vemos que estamos percorrendo as atrações que tínhamos carregadas em nossa base de dados.... mas agora com a aplicação com um desenho mais apropriado.

Não há dúvida que, se dispomos de um designer em nossa equipe, nunca faríamos o trabalho de modificar as classes e propriedades manualmente, mas a importação destas definições a partir do Sketch resolve o desenho de forma mais profissional e podemos nos dedicar à parte funcional de nossa aplicação.

Para concluir, vamos gerar em Android para ver como aparece o resultado final com os dados da base de dados no dispositivo móvel.

Execução da aplicação com dados reais





E também vemos como a aplicação em Android está mostrando todas as atrações que temos na base de dados, e também com o novo desenho.

Esta forma de trabalhar, onde o desenho é parte do desenvolvimento, é refletida na aplicação da metodologia DevOps, que veremos mais adiante.

GeneXus™

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