

# EXERCÍCIOS PRÁTICOS

## Curso GeneXus for Mobile

Departamento de Treinamento - GeneXus  
Janeiro 2024

<b>CONTEÚDO .....</b>	<b>2</b>
O PROBLEMA .....	3
NOVO PROJETO, NOVA BASE DE CONHECIMENTO .....	3
MODELO DE TRANSAÇÕES, OBJETOS DO BACKOFFICE E DADOS DE TESTE .....	4
CONFIGURAÇÃO DE LIVE EDITING .....	5
PRIMEIROS PASSOS E CONTROLES BÁSICOS DE UI .....	5
APLICANDO DESIGN À TELA DE PARQUES DE DIVERSÕES .....	9
CONTROLE CANVAS .....	12
MÚLTIPLOS LAYOUTS EM UM GRID .....	13
MÚLTIPLOS LAYOUTS PARA UM PANEL .....	17
USO DE GRIDS .....	17
PADRÃO WORK WITH .....	20
PROGRAMAÇÃO DE EVENTOS EM UM PANEL .....	22
USO DE API'S .....	23
PANEL COM FUNCIONAMIENTO OFFLINE .....	33
ADICIONANDO SEGURANÇA À APLICAÇÃO .....	35

## O PROBLEMA

Uma multinacional responsável pela gestão de parques de diversões contrata você para que desenvolva um sistema em uma plataforma móvel nativa em Android, para visualizar e eventualmente atualizar a informação com a qual trabalha. Imagine que o sistema é composto por um módulo:

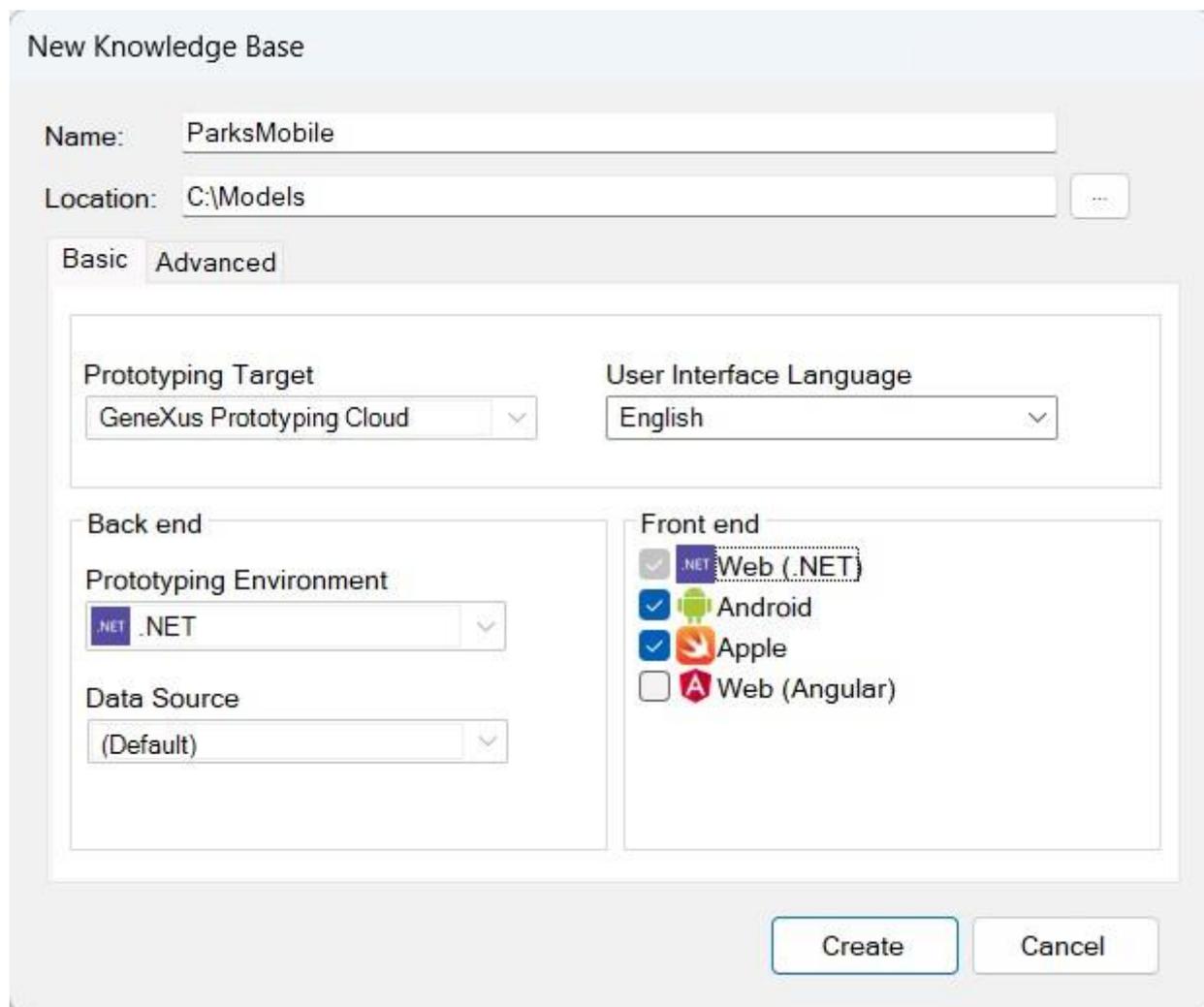
- **Back-end:** parte da aplicação que deverá rodar em um servidor web, composta pelos serviços REST gerados em .NET, que devem poder ser acessados pela aplicação móvel nativa a partir de qualquer lugar com conexão à internet.
- **App mobile (front-end):** parte da aplicação gerada em Android que será executada em um emulador ou em um dispositivo Android nativo, conectada ao servidor web do back-end através de WiFi, de maneira tal que os usuários da aplicação possam acessar a informação a partir de qualquer lugar com conexão à internet.

## NOVO PROJETO, NOVA BASE DE CONHECIMENTO

Entre no GeneXus e crie uma base de conhecimento chamada *ParksMobile\_NomeSobrenome* para começar o desenvolvimento da aplicação. Onde diz "NomeSobrenome" deve ser incluído o nome e sobrenome do aluno que está realizando o prático. Incluir o nome será importante mais adiante quando carregar a KB no GeneXus Server.

### Sugerimos:

- Escolha como ambiente de desenvolvimento .NET. Certifique-se de ter instalado tudo o que é necessário (incluindo o SQL Server). Se você usa GeneXus Trial, o ambiente de geração com .NET e SQL Server já está predefinido, prototipando automaticamente na nuvem de Amazon. Se você usa a versão GeneXus Learning ou GeneXus Full, deverá inserir estes dados ao criar a KB.
- No grupo Front end, marque Android e se você também quiser gerar a aplicação para Apple, pode marcar essa caixa.
- Não crie a base de conhecimento na pasta "Meus Documentos" ou em qualquer outra pasta que fique sob "Documents and Settings", pois essas pastas possuem permissões especiais concedidas pelo Windows.



## MODELO DE TRANSAÇÕES, OBJETOS DO BACKOFFICE E DADOS DE TESTE

Este prático é focado na construção de objetos nativos para uma aplicação móvel, portanto a primeira tarefa é importar um arquivo .xpz que contém as transações junto com os data providers que permitem o preenchimento das tabelas e como todas as transações têm aplicado o padrão Work With for Web, também são incluídos os objetos gerados pelo padrão e outros recursos, de forma a ter o back-office web completo e funcionando.

O arquivo a ser importado está disponível na página do curso, na seção Materiais. Também estão disponíveis as imagens que podem ser necessárias para as entidades do prático.

Após importar o xpz, pressione **F5** para criar a aplicação de back-office web e familiarizar-se com os dados que iremos usar. Se você estiver utilizando GeneXus Learning ou

GeneXus Full, será aberta uma caixa de diálogo onde deverá preencher os dados correspondentes ao nome da base de dados, o nome do servidor e o tipo de conexão.

Depois que for exibida a mensagem de que foi executado com sucesso o Developer Menu, na janela do launchpad execute o objeto WWCountry para visualizar os dados dos países.

À esquerda da tela você tem disponível um menu que permite trabalhar com outras entidades, por exemplo os parques de diversões. Sugerimos que navegue entre as diferentes entidades para ver os seus dados.

Sugestões:

- Mantenha a janela de propriedades do IDE aberta (**F4**), pois a utilizará continuamente.
- Verifique se você tem instalados todos os requisitos necessários do gerador Android e também o Android SDK.
- Assegure-se de que nas propriedades do nó Front End a propriedade Generate Android esteja como True e a propriedade Main Platform tenha o valor Android. Opcionalmente, você pode deixar a propriedade Generate Apple definida como True. O restante das propriedades deixe com seus valores padrão.
- Você pode usar o emulador que GeneXus cria automaticamente ou criar o seu próprio como mostra o vídeo "Primeiros passos com uma aplicação móvel nativa", mas é conveniente que o emulador de Android esteja em execução antes de executar a aplicação nativa.

## CONFIGURAÇÃO DE LIVE EDITING

Antes de começar a criar os objetos, configure o uso de Live Editing para agilizar seu trabalho.

Lembre-se:

- Alterar o combo com o valor **Release** para o valor **Live Editing**.
- Fazer *Rebuild All* para que a alteração se aplique: isso só será necessário na primeira vez.

## PRIMEIROS PASSOS E CONTROLES BÁSICOS DE UI

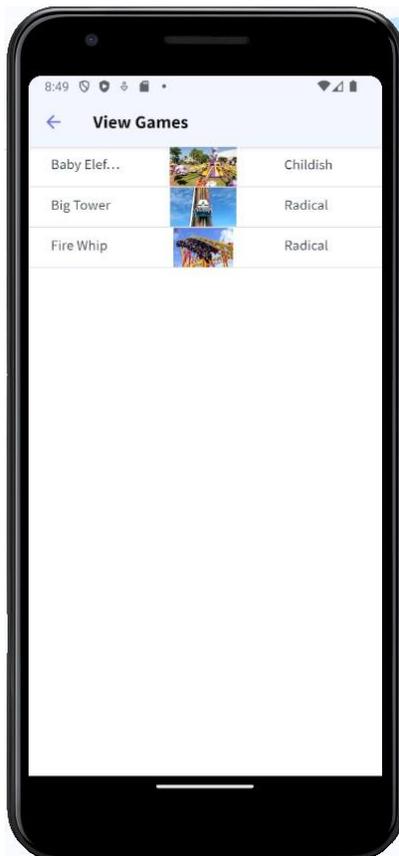
A empresa que administra os parques solicita a implementação de uma tela (objeto Panel) chamada ViewParks que mostre a lista de parques de diversão disponíveis, com informação do identificador, nome, foto, nome de cidade e nome de país ao qual pertence cada um.

A tela deverá ficar parecida com a seguinte:



Ao tocar sobre um dos parques, deverá ser aberta outra tela onde são exibidos seus brinquedos, cada um com os dados correspondentes ao nome, foto e nome de categoria.

Esta tela deve se chamar ViewGames e ser parecida com a seguinte:.



Lembre-se:

- Para ser possível mostrar os jogos do parque selecionado, o panel ViewGames deve receber por parâmetro o identificador do parque de diversões e ele deve ser passado como argumento na invocação a ViewGames a partir de ViewParks.

Para executar a tela ViewParks faremos isso a partir de um objeto menu.

Para isso criamos um objeto Menu que chamaremos de MenuMain. Em seguida, arrastamos o objeto ViewParks para o nó Items do objeto menu.

Vamos às propriedades da action criada e atribuímos a propriedade Image com a imagem Parks, que foi previamente importada no xpz que nos permitiu importar o back-office.

Se formos aos eventos, veremos o evento ViewParks e com a invocação ao panel ViewParks:

```

Menu * | Events * | Variables | Documentation
'ViewParks'
1 | Event 'ViewParks'
2 |     ViewParks()
3 | EndEvent

```

O objeto menu deveria ficar assim:

The screenshot shows the IDE interface with the 'Menu' object selected in the 'Items' list. The 'Properties' panel on the right displays the following information for the selected 'Action (ViewParks)':

Name	<b>ViewParks</b>
Description	<b>View Parks</b>
Image	<b>Parks</b>
Unselected Image	(none)
Class	MenuItem

Lembre-se:

- Configurar o objeto MenuMain como startup, clicando com o botão direito sobre o nome do objeto e selecionando **Set as startup object**. Isso fará com que ao pressionar F5 seja executado o MenuMain no emulador ou dispositivo.

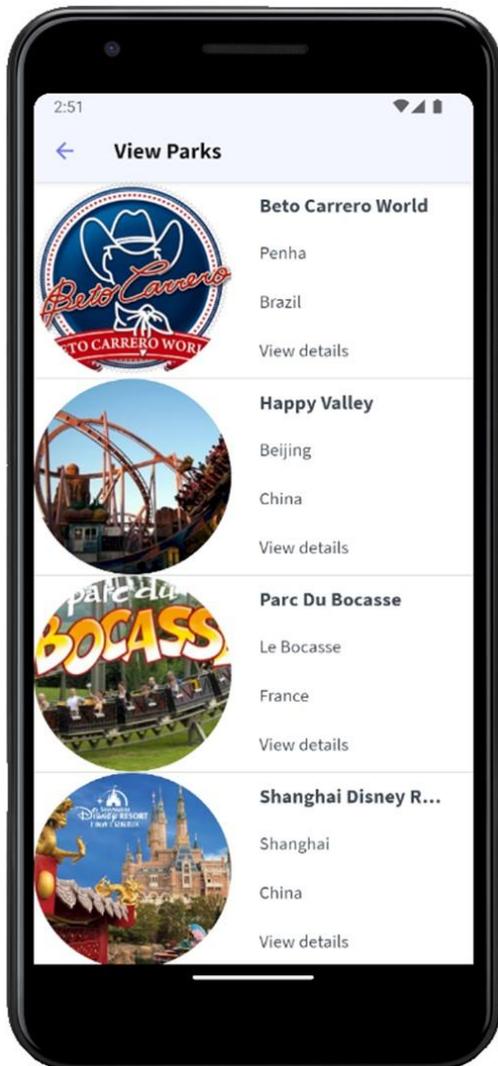
Agora clicamos sobre a aba do objeto MenuMain e vamos para suas propriedades.

Na propriedade **Application Title** escrevemos: Amusement Parks e depois no grupo Android, na propriedade **Android Application Icon** selecionamos a imagem AmusementParksIcon, que também veio no xpz que importamos.

Verifique se você já tem o emulador em execução e pressione **F5** para executar o objeto MenuMain e ver o panel ViewParks em execução. Aproveite para verificar no dispositivo o nome da aplicação e o ícone atribuídos. Verifique também se você vê corretamente o panel ViewGames com os brinquedos do parque selecionado.

## APLICANDO DESIGN À TELA DE PARQUES DE DIVERSÕES

Para a tela de visualização de parques que você acabou de criar, adicione design e altere a apresentação dos atributos no Grid para que fique da seguinte maneira:



Isto implica que a foto do parque esteja à esquerda e restante dos atributos à direita, ligeiramente separados da foto, que a foto ocupe todas as linhas do grid, que seja redonda e que o nome do parque apareça em negrito.

### Sugestões:

- Crie um novo objeto do tipo Design System chamado ParksMobileDSO que herde de UnanimoMobile com as classes necessárias para alterar a aparência dos controles.
- Use propriedades como **border-radius** e **gx-content-mode** para realizar as alterações necessárias. Como orientação, para que a foto fique redonda, você deve usar um valor em pixels para border-radius, que seja a metade do valor dos dips atribuídos à coluna da foto (por exemplo, se para a coluna você deu 180 dips de largura, então border-radius deve ser 90px).

- Para mostrar o título em negrito, abra o Design System Object Unanimomobile para ver como está definida a classe que utilizam os atributos. Copie todas as propriedades para uma nova classe e altere o valor da propriedade font-family de `$fonts.primary-regular` para `$fonts.primary-bold`.

Não esqueça:

- Configurar o Design System que você acabou de criar para que seja aquele considerado pela plataforma para a qual você está desenvolvendo. Nó **Customization** → **Platforms** • Associar as classes aos controles.
- Embora o identificador do parque não deva ser visto, você precisa dele no grid para passá-lo por parâmetro

Além disso, a empresa solicita outra tela chamada ViewParkDetail com a informação detalhada do parque selecionado pelo usuário ao tocar sobre “View details”:



Esta nova tela deve ficar assim, com o nome do parque destacado, a foto ocupando toda a largura da tela, depois o site web e o endereço do parque com seus respectivos rótulos e um botão com o rótulo BACK:



Ao tocar no botão BACK, deverá retornar à tela anterior (ViewParks).

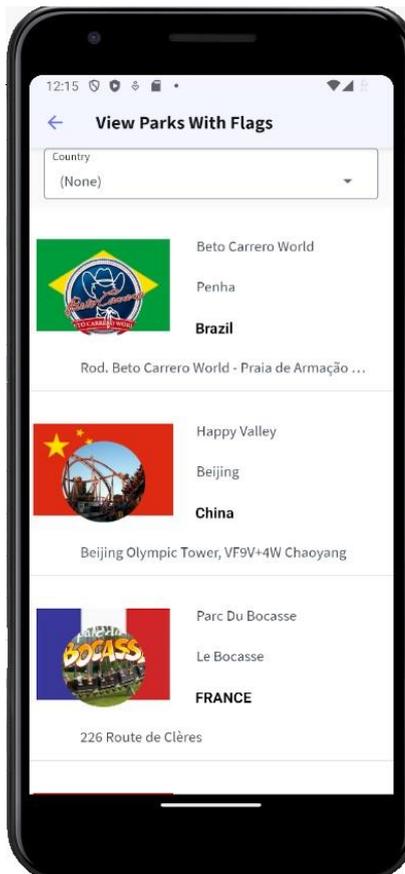
Sugestões:

- Crie uma variável e carregue-a no evento Start com o texto "View details".
- Use tokens de cor para facilitar seu trabalho nos estilos.
- Use propriedades como **color**, **background-color**, **border-radius**, **font-family** e **font-size** para realizar as alterações necessárias.
- Use as propriedades **Columns Style** e **Rows Style** da tabela contêiner para definir o espaço que ocupam os elementos nas linhas e colunas.
- Use as propriedades **Horizontal Alignment** e **Vertical Alignment** para alinhar os elementos nas células.

## CONTROLE CANVAS

É necessário implementar um panel chamado ViewParksWithFlags que mostre os dados básicos dos parques de diversões, mas cada foto do parque deve ser vista com a bandeira de seu país ao fundo.

Esta tela deverá ser parecida com a seguinte:



Além disso, deve ser adicionado acima da tela um combobox que permita selecionar um país e filtrar a lista de parques mostrando apenas aqueles que pertencem ao país selecionado.

Para terminar, arrastamos o panel ViewParksWithFlag para o nó Items do menu MenuMain, atribuindo a propriedade Image com a imagem ParksWithFlags.

Lembre-se:

- O controle **Canvas** é um objeto contêiner do tipo tabela, de modo que você pode arrastar controles para dentro dele e ajustar valores de alinhamento, Row Span ou Column Span, como em uma tabela.

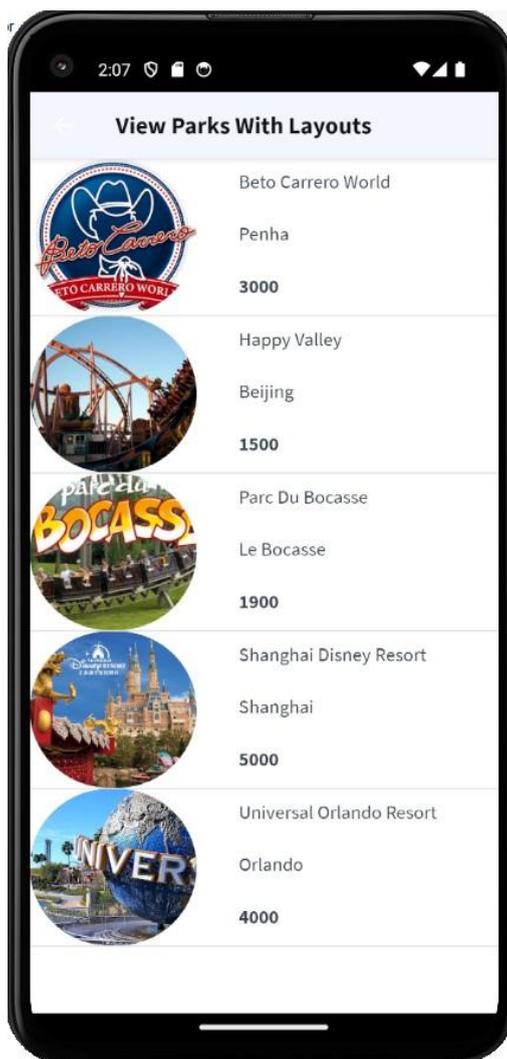
- Ao inserir um controle dentro de um Canvas, automaticamente é adicionada uma propriedade Z-Order que permite definir a camada onde estará localizado o controle. O valor 0 é a camada mais abaixo e quanto maior o número de **Z-Order**, mais acima será a camada onde se localiza o controle, pelo que é possível sobrepor controles uns sobre outros.
- Além da propriedade Z-Order, você deve atribuir o restante das propriedades do grupo **Absolute position**, para que veja corretamente localizado o controle no canvas.
- Para que ao selecionar um valor do dynamic combobox de países tenha efeito na lista de parques, você deve programar o evento ControlValueChanged da variável do dynamic combobox com um comando Refresh e adicionar uma regra Parm com a variável do combo.

## MÚLTIPLOS LAYOUTS EM UM GRID

É desejado alterar a informação que mostra o panel ViewParks, dependendo de determinadas condições. Antes de continuar, faça um Save As deste panel e salve-o como ViewParksWithLayouts.

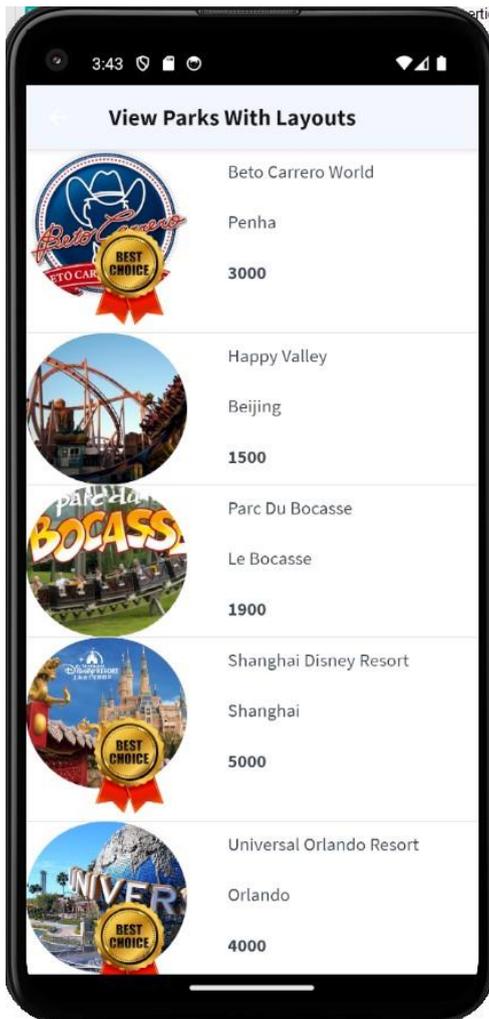
Por padrão é desejado que mostre os parques de diversões disponíveis na base de dados, mostrando a foto, o nome, a cidade e a média diária de visitantes do parque.

A tela com os dados padrão deverá ser parecida com a seguinte:



Para aqueles parques de diversões que a quantidade de visitantes ultrapassa os 2000 diários, automaticamente deve ser exibido o selo de “Melhor escolha” sobreposto à foto do parque. Além disso, para esses parques, o dado da quantidade de visitantes deve ser mostrado na cor vermelha e destacado em negrito. Para o restante, não deve ser mostrado o selo, nem o número de visitantes destacado.

A tela com o selo de “Melhor escolha” sobreposto deve ser vista semelhante a esta:

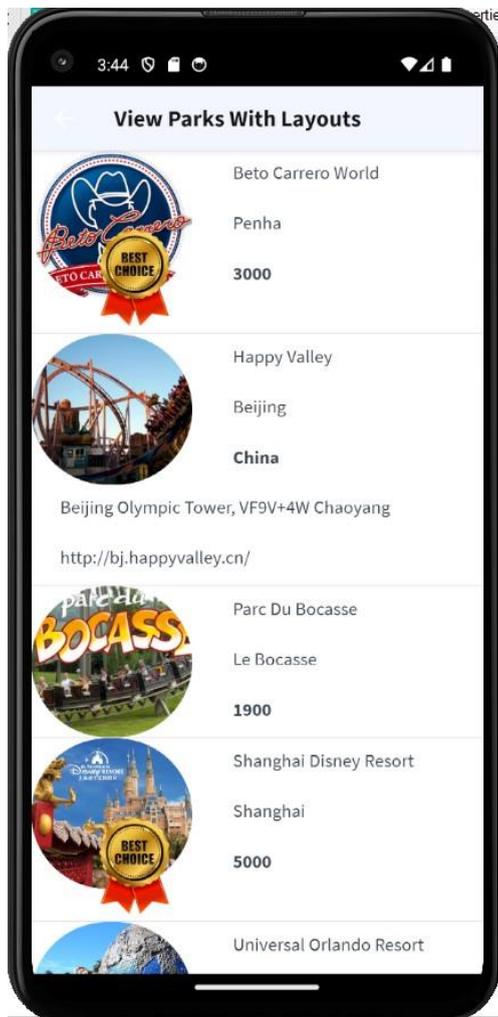


#### Sugestões:

- Para poder atribuir mais de um layout a um panel, você deve primeiro criar um layout novo clicando na seta do canto superior direito da tabela do grid e escolher Add New Item Layout.
- Para que o selo Best Choice se sobreponha à foto da atração, você pode utilizar um **controle canvas**. Para fazer isso, você deverá primeiro remover todos os controles do grid, inserir o canvas no grid e a imagem do selo, depois inserir um controle tabela e inserir os controles novamente.
- Uma forma fácil de fazer isso é em primeiro lugar copiar a tabela do grid, depois apagar os controles da tabela, inserir o canvas com a foto e depois colar a tabela sobre o controle canvas. Por último, atribuímos à imagem de selo um valor de Z-Order maior que o da tabela que contém os controles.
- Não se esqueça de atribuir as propriedades do grupo **Absolute position** da imagem e atribuir uma altura de 100% à tabela.
- Para atribuir um layout dinamicamente, você deve fazê-lo no evento Load do panel, verificando a condição (neste caso que o número de visitantes seja maior que 2000) e atribuindo a propriedade ItemLayout do grid com o nome do layout correspondente.

Além do exposto acima, ao tocar sobre qualquer parte da informação do parque, não deve mais ser invocado o panel que mostra os games, mas deve ser ocultada a quantidade de visitantes e exibido o endereço, o país e a página web.

A tela com mais informação sobre os parques deve ser vista como a seguinte:

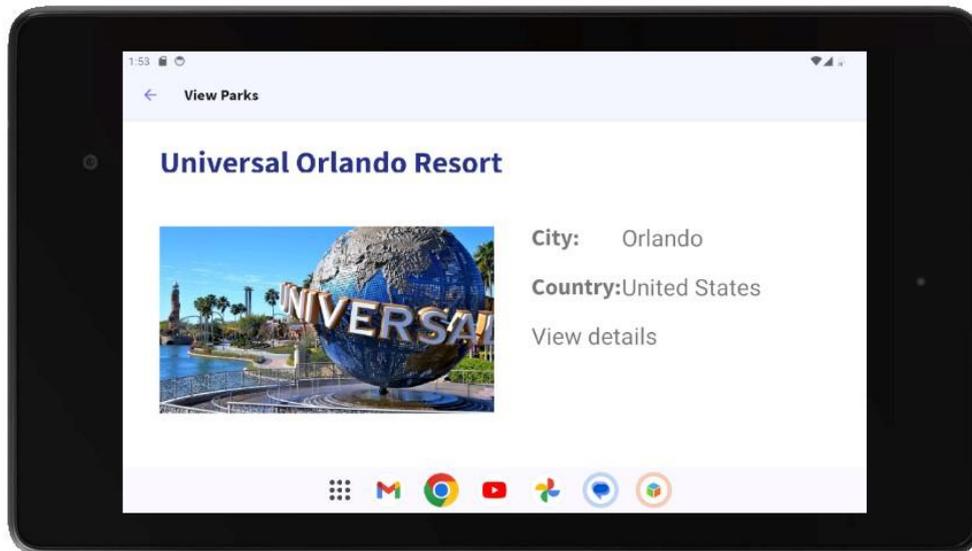


Lembre-se:

- Até agora, estava atribuído o valor da propriedade **Default Action** do grid a um evento que invocava o panel ViewGames. Isso fazia com que ao tocar sobre um dos parques, fosse aberta a tela que mostrava os brinquedos do parque selecionado. Se desejar que ao tocar seja invocado outro layout diferente do layout padrão (Layout1), você deverá voltar a propriedade Default Action ao seu valor padrão e atribuir a propriedade **Default selected item layout** com o nome do layout que mostra o detalhe do parque selecionado.
- Para executar o panel ViewParksWithLayout você pode adicioná-lo como action ao MenuMain e lhe atribuir uma imagem como ícone, ou também pode definir o panel como objeto main e executá-lo diretamente.

## MÚLTIPLOS LAYOUTS PARA UM PANEL

É necessário que a tela ViewParks reajuste seu conteúdo quando a aplicação estiver sendo usada em um Tablet de 7" na orientação horizontal, para que fique parecida com a seguinte:



Sugestões:

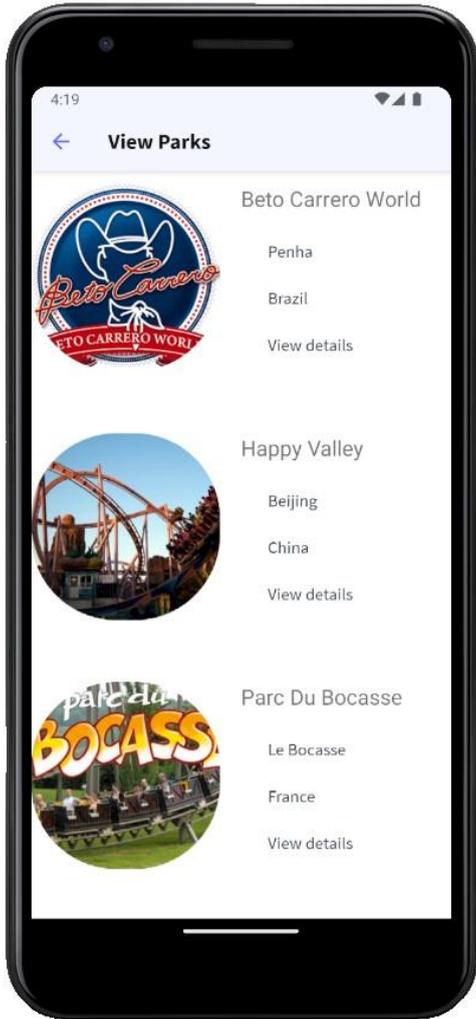
- Crie um novo device no Android Studio para simular um Tablet de 7".
- Não se esqueça de criar um novo layout para o panel.
- Use as propriedades **Columns Style**, **Rows Style** e **Col Span** da(s) tabela(s) contêiner(s) para definir o espaço que ocupam os elementos nas linhas e colunas.
- Use classes com as propriedades que você usou anteriormente para alterar a aparência do nome dos parques e o tamanho da informação mostrada. Para alterar o estilo dos rótulos dos atributos, use a `gx-label-class` e atribua uma classe que altere esse estilo.

Lembre-se:

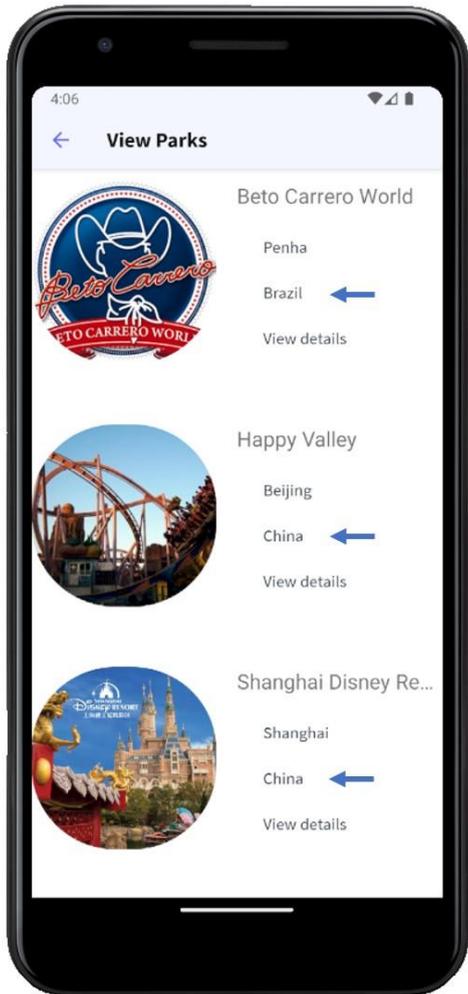
- Para visualizar um único registro por linha e rolar horizontalmente, você pode usar as propriedades **Scroll direction**, **Items Layout Mode** e **Items Per Column** ou alterar o tipo do Grid para **Horizontal**.

## USO DE GRIDS

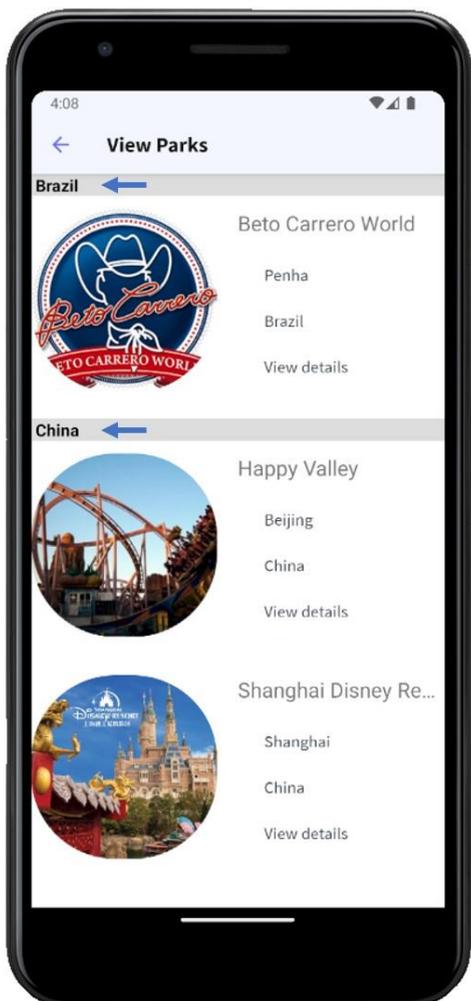
Modifique o layout padrão do panel ViewParks para que o Grid tenha uma rolagem horizontal e mostre no máximo 3 linhas por página. A tela deverá ser parecida com a seguinte:



Agora, os dados deverão sair classificados em ordem crescente por nome de país:



Agora, é necessário que os parques de diversões sejam agrupados por país:



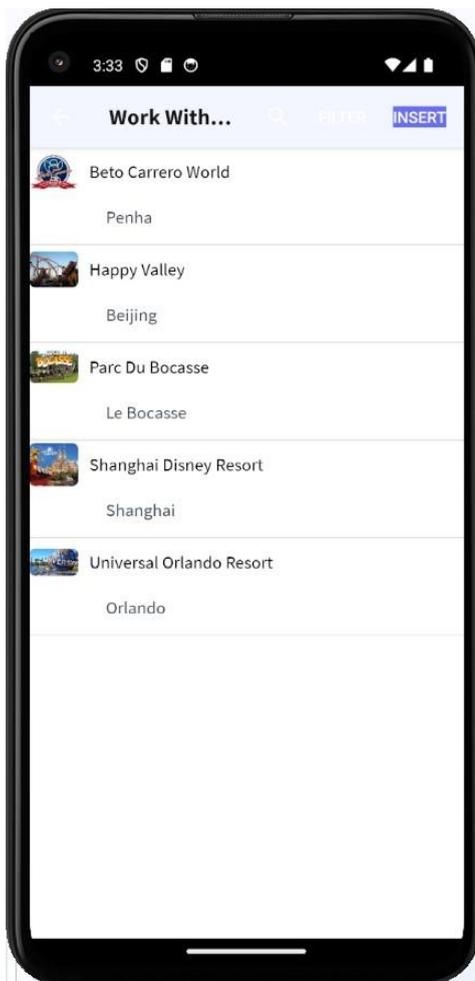
## PADRÃO WORK WITH

Para ser possível realizar ações de CRUD (Criar, ler, atualizar, excluir) sobre os parques de diversões, de uma maneira simples a partir de um dispositivo móvel, é necessário o seguinte:

Aplicar o padrão Work With à transação AmusementPark

Adicioná-lo como item ao objeto Menu MenuMain

Ao acessar a lista de parques de diversões através do work with, é desejado que além do nome e foto de cada parque, apareça o nome da cidade onde se encontra, semelhante ao que é mostrado na seguinte imagem:



Sugerimos que você melhore o design desta tela.

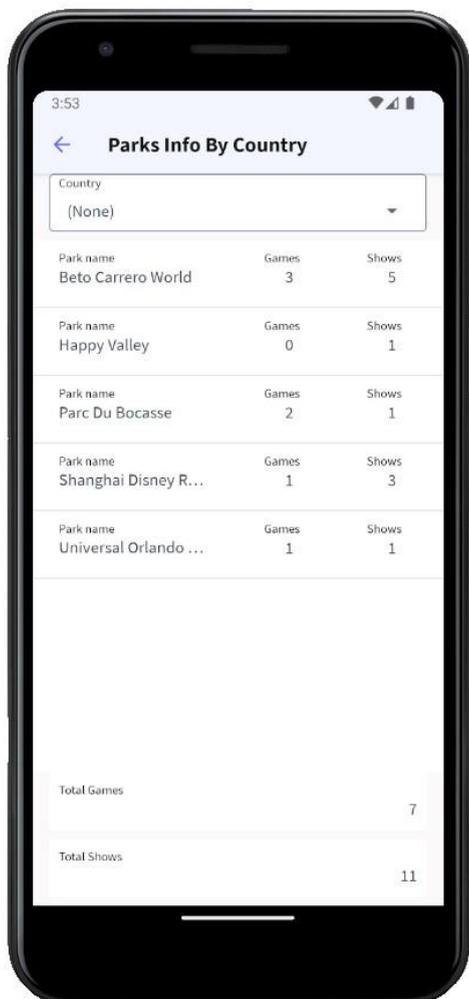
Também é desejado que ao pesquisar, além de ser possível procurar por nome do parque, na mesma search também seja possível pesquisar por nome de cidade. Então, quando pesquisamos por nome de parque, nos traga os parques que correspondam a esse nome, e se pesquisamos por nome de cidade, nos traga os parques que sejam dessa cidade.

#### Sugestões:

- Para mostrar a cidade junto ao parque de diversões, devemos fazê-lo a partir do nó List do padrão WorkWith, adicionando o atributo correspondente
- Para pesquisar por mais de um atributo, deverá ser feita a configuração nas propriedades do grid (que está no nó List) e configurar a propriedade "Search". Adicionando o outro atributo pelo qual queremos pesquisar.

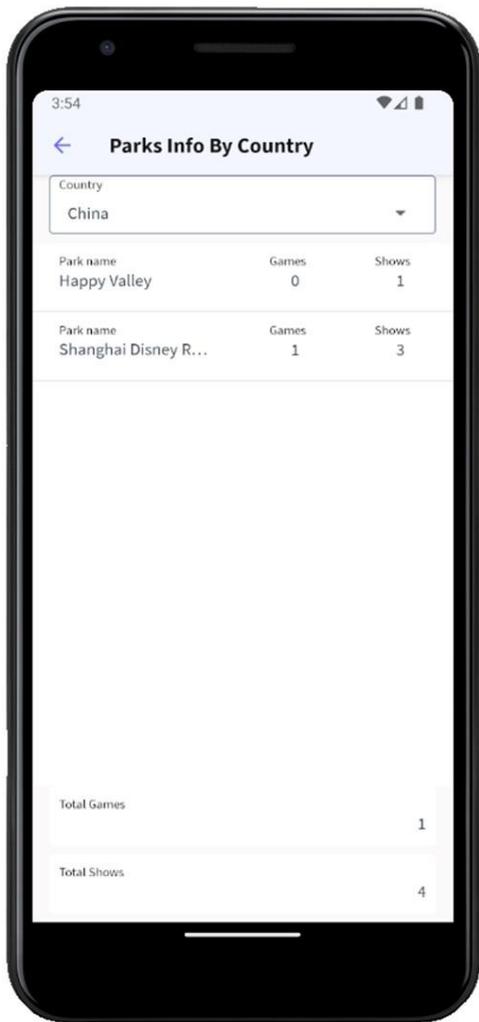
## PROGRAMAÇÃO DE EVENTOS EM UM PANEL

É necessário desenvolver uma tela que mostre para cada parque de diversões o seu total de brinquedos e o seu total de shows, permitindo filtrar por país. Além disso, deve mostrar abaixo dessa lista os totais gerais. A tela deverá ser parecida com a seguinte:



Park name	Games	Shows
Beto Carrero World	3	5
Happy Valley	0	1
Parc Du Bocasse	2	1
Shanghai Disney R...	1	3
Universal Orlando ...	1	1
Total Games		7
Total Shows		11

Filtrando por país:



Adicioná-lo como item ao objeto Menu MenuMain.

Lembre-se:

- Que os Panels são programados de maneira diferente dos WebPanels, pois a parte fixa e a parte variável usam Data Providers diferentes para sua carga.

Sugestões:

- Use os eventos Load e Refresh para carregar as variáveis do Grid e da parte fixa do Panel respectivamente.

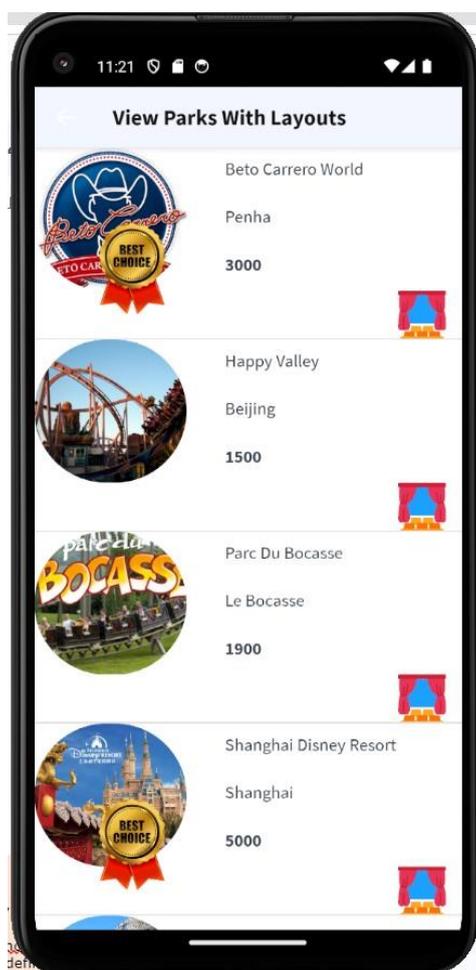
## USO DE API's

Parte 1)

É necessário invocar uma API que permita criar um compromisso no calendário do dispositivo, de forma a agendar um evento para um show previamente selecionado.

Antes de continuar, é necessário poder ver a lista de shows de um parque de diversões e depois visualizar o detalhe do show que queremos ver. Para isso, no panel ViewParksWithLayouts deve ser mostrada para cada parque, uma imagem que simbolize os shows, para que ao clicar sobre ela nos leve à lista de shows desse parque.

O ícone de shows em ViewParksWithLayouts pode ser visto desta maneira:

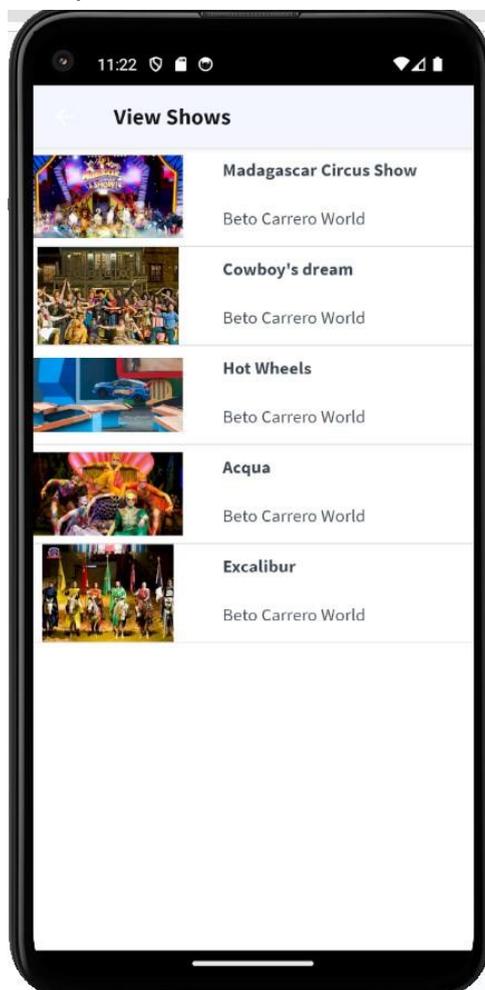


#### Sugestão:

- Crie uma variável &Shows do tipo Image e no evento Load do grid atribua a imagem dos shows, com o método FromImage da variável.
- Arraste uma tabela para a tabela do grid e dentro dela coloque a variável &Shows com a propriedade Label position definida como None. Você deveria repetir isso para todos os layouts que definiu que deseja que acessem os shows.

Ao clicar sobre o ícone de shows de um parque, deve ser aberto o panel ViewShows com a lista de shows que o parque possui, mostrando a foto e nome de cada show, e o parque selecionado.

Este panel deve ser semelhante ao seguinte:



Então, ao clicar sobre um show, deverá ser aberto o detalhe do show invocado, para o qual criaremos um panel chamado ShowMoreInfo, com a foto ampliada do show e os dados do nome do show, a hora de início e término, a cidade e o país.

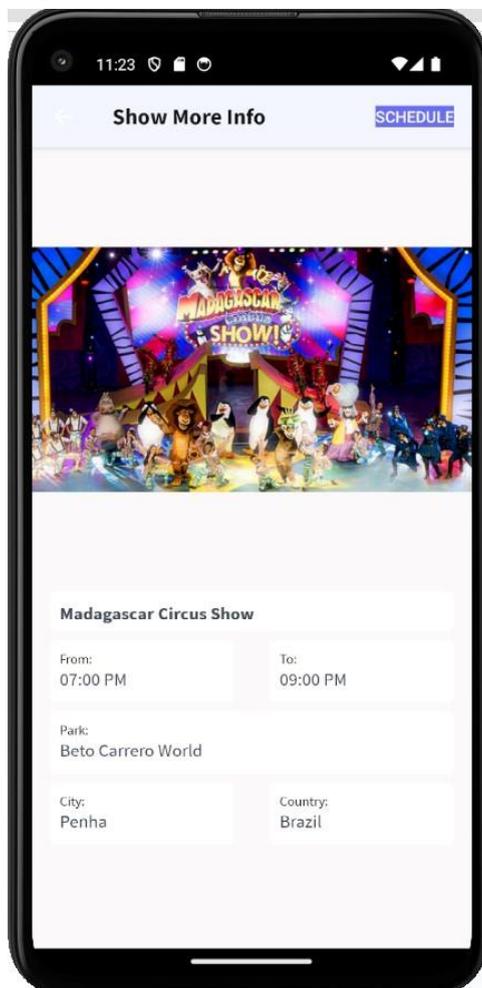
Para criar um compromisso no calendário que nos lembre de assistir à apresentação deste show, adicionamos um botão com o texto SCHEDULE, em cujo evento invocamos um panel que nos permita inserir a data que queremos ir ao show.

Lembre-se:

- No panel ShowMoreInfo, dar o design adequado à tabela que contém os controles, posicionando-os de forma que sejam vistos como a tela de exemplo. Além disso, não se esqueça de atribuir valores adequados às propriedades Column Style e Row Style da tabela para visualizar o detalhe do Show com a foto ampliada e os dados legíveis.

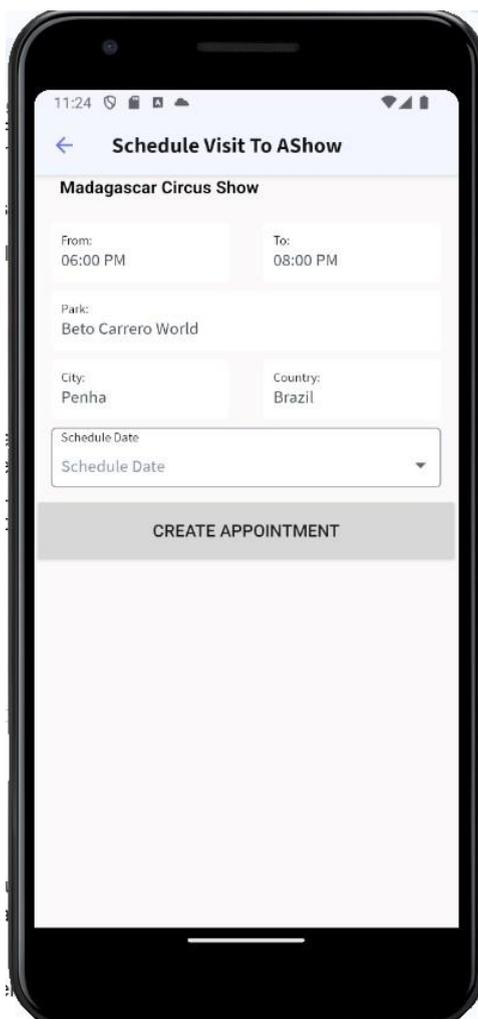
- O botão SCHEDULE deve ser inserido no controle Application Bar do panel, clicando com o botão direito sobre ele e selecionando Insert button...

O panel ShowMoreInfo com a informação do show e o botão SCHEDULE deveria ficar da seguinte forma:



O panel invocado pelo botão SCHEDULE devemos criar com o nome ScheduleVisitToAShow e nos mostrará os dados do show, incluindo os horários de início e término da apresentação, e deverá incluir a possibilidade de inserir a data que escolhemos para a apresentação, já que todos todos os dias há apresentação no mesmo horário.

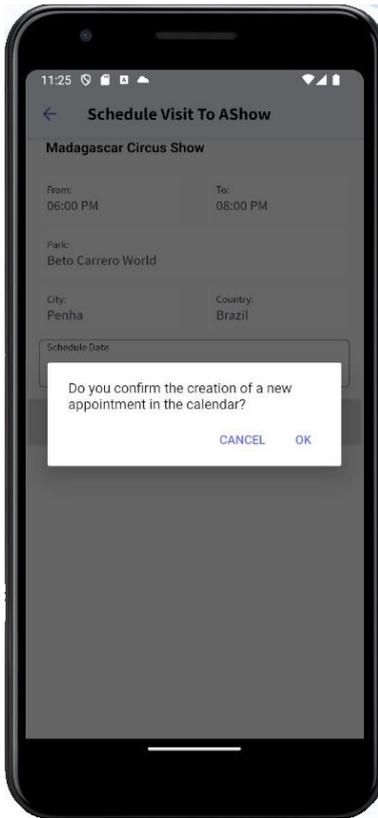
A tela do panel ScheduleVisitToAShow deverá ser semelhante à seguinte:



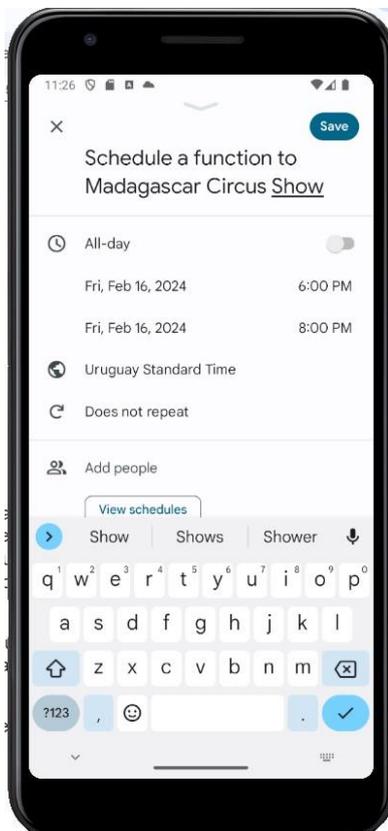
Então, com esses dados, deverá ser invocada uma API que permita adicionar uma entrada no calendário nativo do dispositivo. A invocação deve incluir um texto que indique que será feito um agendamento para o show selecionado (por exemplo “Schedule a function to Madagascar Circus Show”), passando então como parâmetros os dados necessários, que dispomos do show.

Antes de ser executada a invocação, deverá ser solicitado ao usuário que confirme a criação do novo evento no calendário, dando a possibilidade de confirmar ou recusar a criação do compromisso do calendário.

O pop-up com a caixa de diálogo para confirmar a ação de criação do compromisso no calendário deve ter a seguinte aparência:



e ao pressionar o botão Confirmar, deverá ser visto o compromisso do calendário a ser salvo.



Lembre-se:

- Para fazer com que apareça um diálogo de confirmação, você deve invocar o método Confirm da API Interop, do nó References, módulo GeneXus.SD, passando como parâmetro a mensagem correspondente.
- Para invocar uma API que adicione um compromisso ao calendário do dispositivo, você deve utilizar o método Schedule da API Calendar, localizada no módulo GeneXus.SD, passando por parâmetro o texto do compromisso, a data de início, data de término, hora de início, hora de término e um texto com o local do agendamento.

## Parte 2)

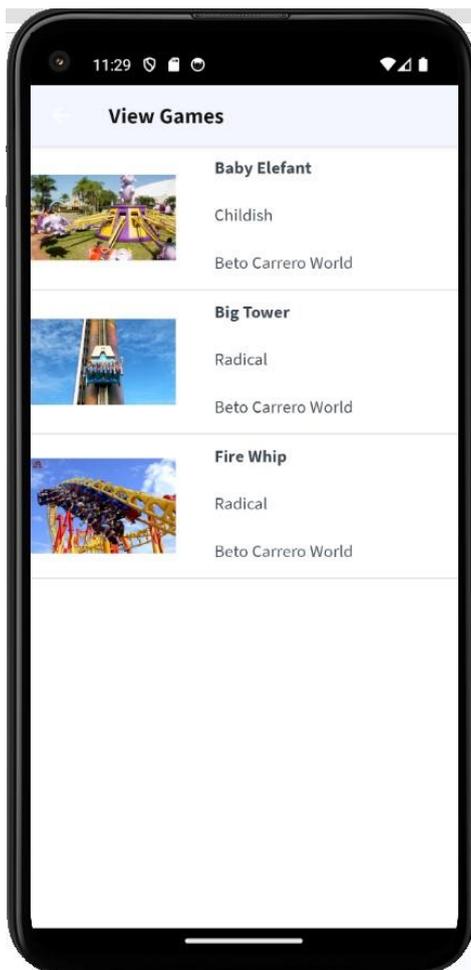
Como parte da promoção dos parques de diversões, é desejado fazer propaganda de um brinquedo de um parque, através de uma mensagem na plataforma X (Twitter), mostrando na mensagem um texto com um máximo de 280 caracteres. O texto da mensagem deve incluir um convite para visitar o brinquedo, que mencionará o nome do brinquedo, o parque ao qual pertence, a cidade e o país, por exemplo: “Enjoy the game Hulk at Shanghai Disney Resort amusement park located in Shanghai, China”.

Para implementar isso, deverá ser adicionado ao panel ViewParksWithLayouts um ícone para ver os brinquedos desse parque, de forma análoga a quando adicionamos o ícone para ver os shows.

A tela ViewParksWithLayouts deverá ficar como segue:

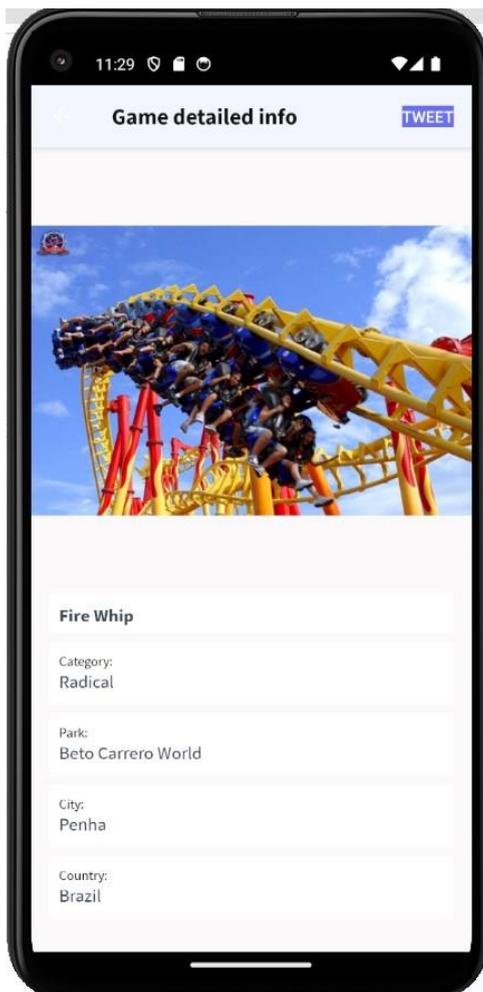


Ao pressionar o ícone de brinquedos, deverá ser invocado o panel ViewGames, que deve ser visto como mostrado a seguir (adicione o design adequado para isso):



Ao clicar sobre um brinquedo, deverá ser aberto um painel que chamaremos de GameMoreInfo com o detalhe do brinquedo, que mostrará a imagem do brinquedo ampliada e abaixo dela o nome do brinquedo, sua categoria, o parque ao qual pertence, a cidade e o país. Também será adicionado um botão TWEET na Application Bar, que executará a ação de publicar o tweet.

A tela do painel GameMoreInfo deve ser parecida com esta:

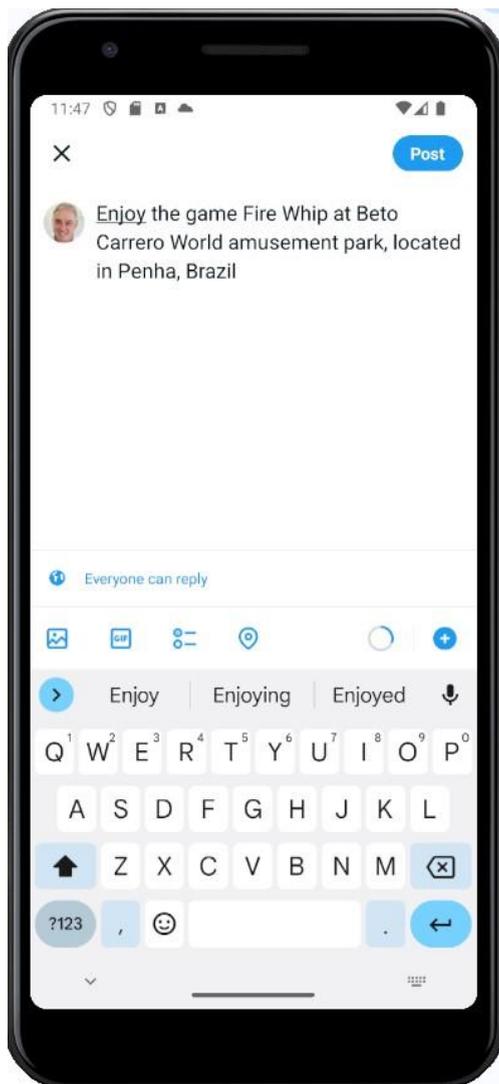


#### Sugestões:

- No panel GameMoreInfo, dar o design adequado à tabela que contém os controles, posicionando-os de forma que sejam vistos como a tela de exemplo. Além disso, não se esqueça de atribuir valores adequados às propriedades Column Style e Row Style da tabela para visualizar o detalhe do Game com a foto ampliada e os dados legíveis.
- O botão TWEET deve ser inserido no controle Application Bar do panel, clicando com o botão direito sobre ele e selecionando Insert button...

Ao pressionar o botão TWEET, deve ser invocada a API que publica o tweet, passando como parâmetro o texto mencionado anteriormente.

Se tiver a aplicação X instalada no dispositivo, ela será aberta mostrando a mensagem a ser publicada, caso contrário será aberto o browser do dispositivo posicionado na página de X, mostrando o tweet a ser publicado.



Lembre-se:

- Para invocar uma API que publique um Tweet, você deve utilizar a API Twitter, localizada no módulo GeneXus.Social, passando por parâmetro o texto do tweet.
- Se você tiver a aplicação X instalada no emulador será aberta a aplicação, caso contrário, será aberta a página do navegador posicionada no site de X. Essas ações podem exigir que você faça login na plataforma para continuar.

## PANEL COM FUNCIONAMIENTO OFFLINE

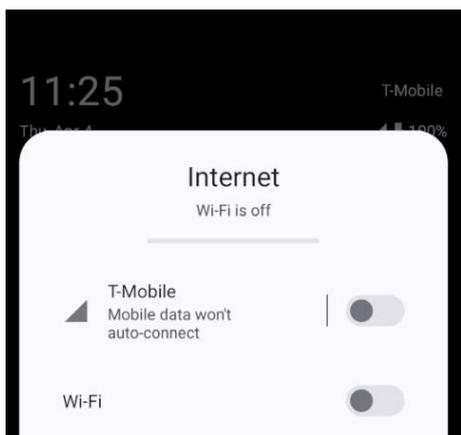
Nos interessa que toda a aplicação também possa ser utilizada de maneira offline.

Portanto, ao ser acessada com conexão à internet, a aplicação se conectará ao server e será sincronizada com a base de dados. Mas se for acessada sem conexão com a Internet, queremos que da mesma forma seja possível navegar sem problemas pelas diferentes telas.

Sugestões:

- Para atender ao requisito, você deverá configurar a propriedade Connectivity Support do objeto MenuMain com o valor Offline.

Após realizar o procedimento correspondente, para provar que efetivamente temos a aplicação funcionando de maneira offline, desconecte a aplicação da internet (desligue Wifi e dados móveis)



Entre no WorkWith AmusementPark e adicione um parque de diversões novo.

Para confirmar que esta alteração tenha sido realizada apenas na base de dados do dispositivo, confirme na interface web, acessando o WWAmusementPark através do Launchpad (View -> Other Tool Windows -> Launchpad), que o registro que você acabou de inserir ainda não aparece.

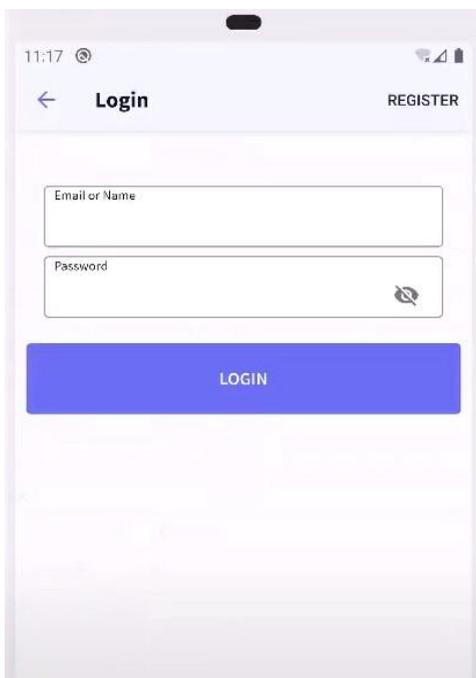
Em seguida, reconecte o dispositivo móvel à internet, feche e abra a aplicação para que seja realizada a sincronização dos dados com o server e então, entre novamente na interface web do WWAmusementPark através do Launchpad. Agora deve aparecer o novo parque de diversões que você inseriu estando offline.

## ADICIONANDO SEGURANÇA À APLICAÇÃO

Aplicar GAM à aplicação, para que ao executar qualquer objeto main, nos peça um login de autenticação. Neste caso iremos testá-lo acessando o objeto menu MenuMain.

Lembre-se:

- Para aplicar o GAM, você deverá definir a propriedade Enable Integrated Security como True, no nó da Versão da KB, no KB explorer.
- Para poder fazer o login na tela gerada pelo GAM, é importante que o dispositivo móvel tenha conexão com a Internet, mesmo que a app continue sendo offline.



Para ser possível implementar o que vem a seguir, precisamos voltar a ter uma aplicação Online. Portanto, você terá que configurar novamente a aplicação nesse modo.

Sugestões:

- Para atender ao requisito, você deverá configurar a propriedade Connectivity Support do objeto MenuMain com o valor Online.

É necessário criar um usuário (com o nome e senha que você escolher), a partir do backoffice web do GAM, que ao efetuar login só tenha permissão para acessar o objeto

WorkWithAmusementPark.

Execute novamente a aplicação e faça login com esse novo usuário para provar que efetivamente você só pode acessar o objeto WorkWithAmusementPark e não o restante dos objetos adicionados ao MenuMain

Sugestões:

- Para criar um novo usuário de login, você deverá entrar no KB Explorer ir até a pasta GAM\_Examples e depois executar (run) o objeto GAMHome
  - No nó da base de conhecimento, você deverá alterar a propriedade Integrated Security para o valor Authorization, para que além de autenticar, controle as permissões.
-



[www.genexus.com](http://www.genexus.com)

Copyright GeneXus S.A. 1988-2024.

Todos los derechos reservados. Este documento no puede reproducirse de ninguna manera sin el permiso expreso de GeneXus S.A. La información aquí contenida está destinada únicamente para uso personal.

Marcas comerciales registradas: GeneXus es marca comercial o marca comercial registrada de GeneXus S.A. Todas las demás marcas comerciales mencionadas en este documento son propiedad de sus respectivos dueños.