

EJERCICIOS PRÁCTICOS

Curso GeneXus for Mobile

Departamento de Capacitación - GeneXus
Enero 2024

CONTENIDO

CONTENIDO.....	2
EL PROBLEMA.....	3
NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO.....	3
MODELO DE TRANSACCIONES, OBJETOS DEL BACKOFFICE Y DATOS DE PRUEBA.....	4
CONFIGURACIÓN DE LIVE EDITING.....	5
PRIMEROS PASOS Y CONTROLES BÁSICOS DE UI.....	5
APLICANDO DISEÑO A LA PANTALLA DE PARQUES DE DIVERSIONES.....	8
CONTROL CANVAS.....	11
MÚLTIPLES LAYOUTS EN UN GRID.....	12
MÚLTIPLES LAYOUTS PARA UN PANEL.....	14
USO DE GRILLAS.....	15
PATRÓN WORK WITH.....	18
PROGRAMACIÓN DE EVENTOS EN UN PANEL.....	19
USO DE API's.....	21
PANEL CON FUNCIONAMIENTO OFFLINE.....	30
AGREGANDO SEGURIDAD A LA APLICACIÓN.....	30

EL PROBLEMA

Una multinacional encargada de gestionar parques de diversiones te contrata para que desarrolles un sistema en una plataforma móvil nativa en Android, para visualizar y eventualmente actualizar la información con la que trabaja. Imagina que el sistema se compone de un módulo:

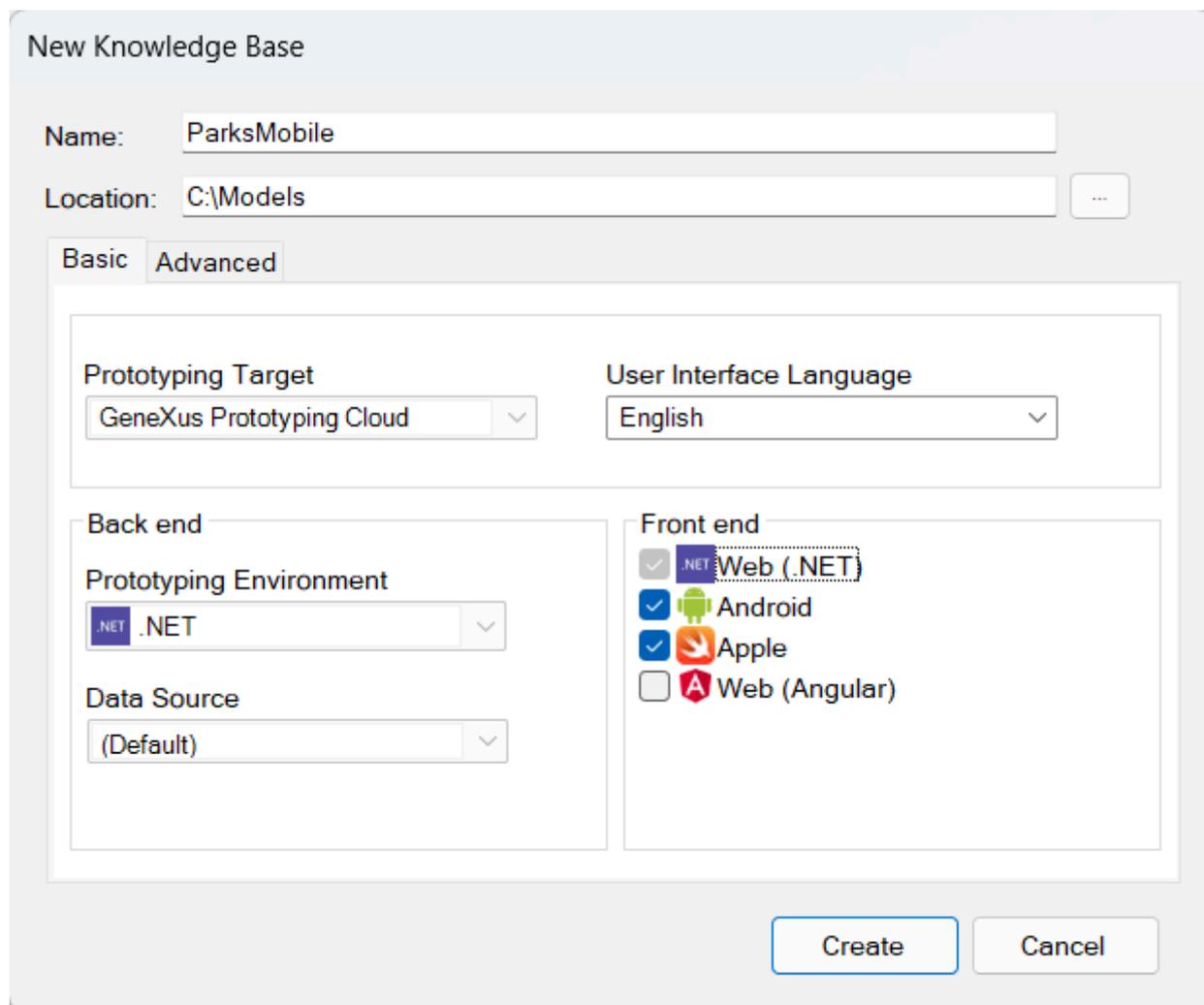
- **Back-end:** parte de la aplicación que deberá correr en un servidor web, compuesta por los servicios REST generados en .NET, que deben poder ser accedidos por la aplicación móvil nativa desde cualquier lugar con conexión a internet.
- **App mobile (front-end):** parte de la aplicación generada en Android que se ejecutará en un emulador o en un dispositivo Android nativo, conectada al servidor web del back-end a través de WiFi, de manera tal que los usuarios de la aplicación puedan acceder a la información desde cualquier lugar con conexión a internet.

NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO

Entra a GeneXus y crea una base de conocimiento de nombre *ParksMobile_NombreApellido* para comenzar el desarrollo de la aplicación. Donde dice "NombreApellido" debe incluirse el nombre y apellido del alumno que está realizando el práctico. Incluir el nombre será importante más adelante cuando se suba la KB a GeneXus Server.

Sugerimos:

- Elige como ambiente de desarrollo .NET. Asegúrate de tener instalado todo lo necesario (incluyendo SQL Server). Si usas GeneXus Trial, el ambiente de generación con .NET y SQL Server ya está predefinido, prototipando automáticamente en la nube de Amazon. Si usas la versión GeneXus Learning o GeneXus Full, deberás ingresar estos datos al crear la KB.
- En el grupo Front end, marca Android y si también deseas generar la aplicación para Apple, puedes marcar esa casilla.
- No crees la base de conocimiento en la carpeta "Mis Documentos" o cualquier otra carpeta que quede bajo "Documents and Settings", debido a que estas carpetas tienen permisos especiales otorgados por Windows.



MODELO DE TRANSACCIONES, OBJETOS DEL BACKOFFICE Y DATOS DE PRUEBA

Este práctico está enfocado en la construcción de objetos nativos para una aplicación móvil, por lo que la primer tarea es importar un archivo .xpz que contiene las transacciones junto a los data providers que permiten poblar las tablas y como las transacciones tienen todas aplicadas el patrón Work With for Web, también se incluyen los objetos generados por el patrón y otros recursos, de forma de contar con el back-office web completo y funcionando.

El archivo a importar está disponible en la página del curso, bajo la sección Materiales. También están disponibles las imágenes que pueden ser necesarias para las entidades del práctico.

Luego de importar el xpz, presiona **F5** para crear la aplicación de back-office web y familiarizarte con los datos que vamos a usar. Si estás utilizando GeneXus Learning o GeneXus Full, se abrirá un cuadro de diálogo donde deberás llenar los datos

correspondientes al nombre de la base de datos, el nombre del servidor y el tipo de conexión.

Luego de que se muestre el mensaje de que se ejecutó exitosamente el Developer Menu, en la ventana del launchpad ejecuta el objeto WWCountry para ver los datos de los países.

A la izquierda de la pantalla tienes disponible un menú que te permite trabajar con otras entidades, por ejemplo los parques de diversiones. Te sugerimos navegar entre las distintas entidades para ver sus datos.

Sugerencias:

- Mantén la ventana de propiedades del IDE abierta (**F4**), pues la utilizarás continuamente.
- Verifica que tengas instalados todos los requerimientos necesarios del generador Android y también el Android SDK.
- Asegúrate de que en las propiedades del nodo Front End, la propiedad Generate Android esté en True y la propiedad Main Platform tenga el valor Android. Opcionalmente puedes dejar la propiedad Generate Apple en True. El resto de las propiedades déjalas con sus valores por defecto.
- Puedes usar el emulador que crea GeneXus automáticamente, o crearte uno propio como muestra el video "Primeros pasos con una aplicación móvil nativa", pero es conveniente que el emulador de Android esté en ejecución antes de ejecutar la aplicación nativa.

CONFIGURACIÓN DE LIVE EDITING

Antes de comenzar a crear los objetos, configura el uso de Live Editing para agilizar tu trabajo.

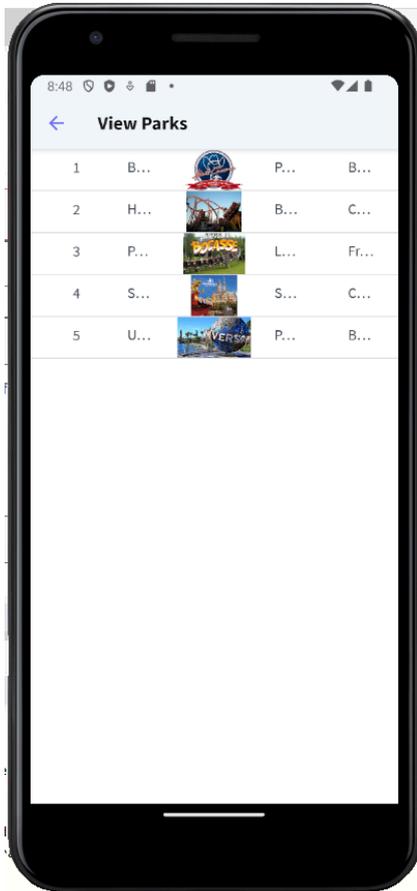
Recuerda:

- Cambiar el combo con el valor **Release** para el valor **Live Editing**.
- Hacer *Rebuild All* para que el cambio se aplique: esto solo será necesario la primera vez.

PRIMEROS PASOS Y CONTROLES BÁSICOS DE UI

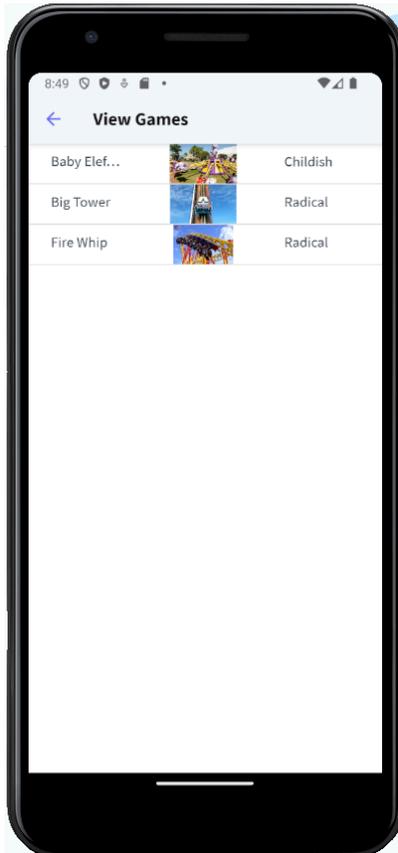
La empresa que gestiona los parques solicita la implementación de una pantalla (objeto Panel) de nombre ViewParks que muestre la lista de parques de diversión disponibles, con información del identificador, nombre, foto, nombre de ciudad y nombre de país al que pertenece cada uno.

La pantalla deberá lucir como la siguiente:



Al hacer tap sobre uno de los parques, se deberá abrir otra pantalla en donde se muestran sus juegos, cada uno con los datos correspondientes a nombre, foto y nombre de categoría.

Esta pantalla deberá llamarse ViewGames y lucir parecida a la siguiente:.



Recuerda:

- Para poder mostrar los juegos del parque seleccionado, el panel ViewGames debe recibir por parámetro al identificador del parque de diversiones y el mismo deberá ser pasado como argumento en la invocación a ViewGames desde ViewParks.

Para ejecutar la pantalla ViewParks vamos a hacerlo desde un objeto menú.

Para eso creamos un objeto Menu al que llamaremos MenuMain. Luego arrastramos al objeto ViewParks al nodo Items del objeto menú.

Vamos a las propiedades del action creado y asignamos la propiedad Image con la imagen Parks, que ya fue previamente importada en el xpz que nos permitió importar el back-office.

Si vamos a los eventos, veremos el evento ViewParks y con la invocación al panel ViewParks:

```

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

```

El objeto menú debería quedar así:

The screenshot shows the IDE interface with the Properties panel open for an Action named 'ViewParks'. The Properties panel contains the following table:

Name	ViewParks
Description	View Parks
Image	Parks
Unselected Image	(none)
Class	MenuItem

Recuerda:

- Setear al objeto MenuMain como startup, dando botón derecho sobre el nombre del objeto y seleccionando **Set as startup object**. Esto hará que al dar F5, se ejecute el MenuMain en el emulador o dispositivo.

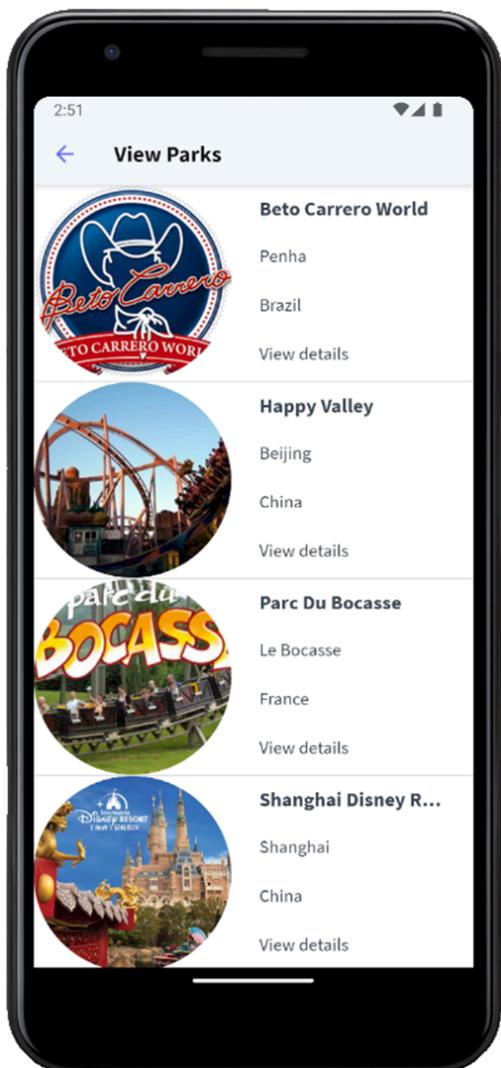
Ahora hagamos clic sobre la solapa del objeto MenuMain y vayamos a sus propiedades.

En la propiedad **Application Title** escribimos: Amusement Parks y luego en el grupo Android, en la propiedad **Android Application Icon** seleccionamos a la imagen AmusementParksIcon, que vino también en el xpx que importamos.

Verifica que ya tienes el emulador en ejecución y presiona **F5** para ejecutar el objeto MenuMain y ver el panel ViewParks en ejecución. Aprovecha para verificar en el dispositivo, el nombre de la aplicación y el icono asignados. Verifica también que veas correctamente el panel ViewGames con los juegos del parque seleccionado.

APLICANDO DISEÑO A LA PANTALLA DE PARQUES DE DIVERSIONES

A la pantalla de visualización de parques que acabas de crear, agrégale diseño y cambia la presentación de los atributos en el Grid para que luzca de la siguiente manera:



Esto implica que la foto del parque esté a la izquierda y el resto de los atributos a la derecha ligeramente separados de la foto, que la foto ocupe todas las filas del grid, que sea redonda y que el nombre del parque aparezca en negrita.

Sugerencias:

- Crea un nuevo objeto de tipo Design System llamado ParksMobileDSO que herede de UnanimoMobile con las clases necesarias para cambiar la apariencia de los controles.
- Usa propiedades tales como **border-radius** y **gx-content-mode** para llevar a cabo los cambios requeridos. Como una guía, para que la foto quede redonda, debes usar un valor en píxeles para border-radius, que sea la mitad del valor de los dips asignados a la columna de la foto (por ejemplo si a la columna le diste 180 dips de ancho, entonces border-radius debería ser 90px).
- Para mostrar el título en negrita, abre el Design System Object UnanimoMobile para ver cómo está definida la clase que utilizan los atributos. Copia todas las propiedades a una nueva clase y cambia el valor de la propiedad font-family de `$fonts.primary-regular` para `$fonts.primary-bold`.

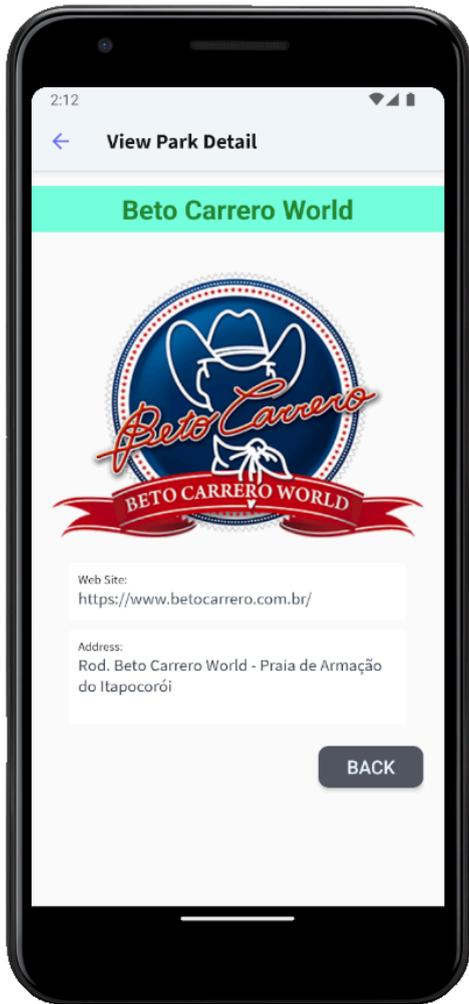
No olvides:

- Configurar el Design System que acabas de crear para que sea el considerado por la plataforma para la que estás desarrollando. Nodo **Customization** → **Platforms**
- Asociar las clases a los controles.
- Que el identificador del parque aunque no debe verse, lo necesitas en el grid para pasarlo por parámetro

Además, la empresa solicita otra pantalla de nombre ViewParkDetail con la información detallada del parque seleccionado por el usuario al hacer tap sobre “View details”:



Esta nueva pantalla debe lucir así, con el nombre del parque resaltado, la foto que ocupe todo el ancho de la pantalla, luego el sitio web y la dirección del parque con sus respectivas etiquetas y un botón con la etiqueta BACK:



Al hacer tap en el botón BACK, se debe volver a la pantalla anterior (ViewParks).

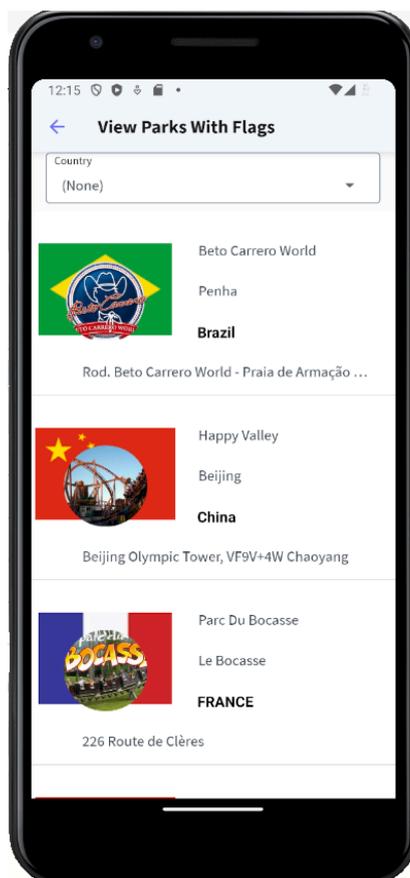
Sugerencias:

- Crea una variable y cárgala en el evento Start con el texto "View details".
- Usa tokens de color para facilitar tu trabajo en los estilos.
- Usa propiedades tales como **color**, **background-color**, **border-radius**, **font-family** y **font-size** para llevar a cabo los cambios requeridos.
- Usa las propiedades **Columns Style** y **Rows Style** de la tabla contenedora para definir el espacio que ocupan los elementos en las filas y columnas.
- Usa las propiedades **Horizontal Alignment** y **Vertical Alignment** para alinear los elementos en las celdas.

CONTROL CANVAS

Se requiere implementar un panel de nombre ViewParksWithFlags que muestre los datos básicos de los parques de diversiones, pero cada foto del parque debe verse con la bandera de su país de fondo.

Esta pantalla deberá lucir parecida a la siguiente:



Además debe agregarse arriba de la pantalla un combobox que permita seleccionar un país y filtrar la lista de parques mostrando únicamente a los que pertenecen al país seleccionado.

Para terminar, arrastramos el panel ViewParksWithFlag al nodo Items del menú MenuMain, asignando la propiedad Image con la imagen ParksWithFlags.

Recuerda:

- El control **Canvas** es un objeto contenedor del tipo tabla, de modo que puedes arrastrar controles dentro del mismo y ajustar valores de alineación, Row Span o Column Span, igual que en una tabla.
- Al insertar un control dentro de un Canvas, automáticamente se agrega una propiedad Z-Order que permite definir la capa donde estará ubicado el control. El valor 0 es la capa de más abajo y cuanto

mayor el número de **Z-Order** más arriba será la capa donde se ubica el control, por lo que es posible superponer controles unos sobre otros.

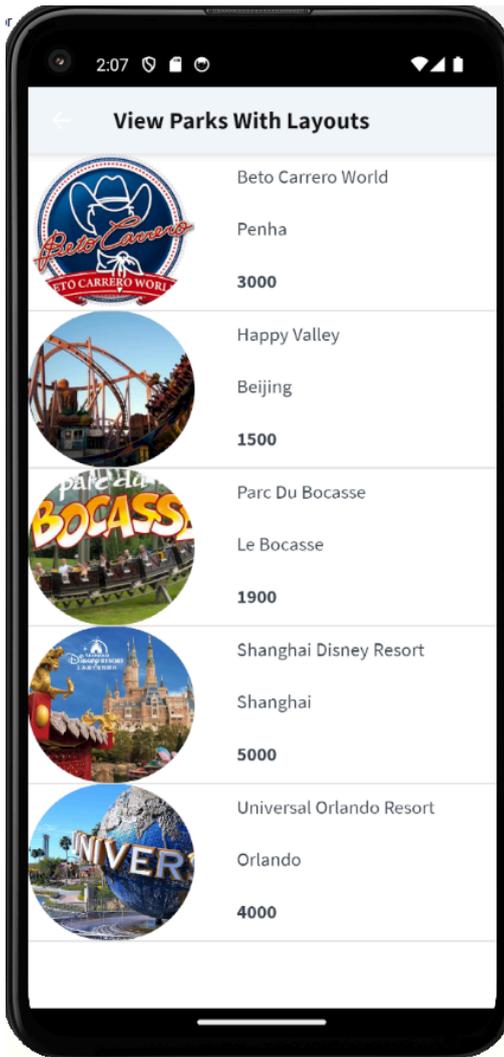
- Además de la propiedad Z-Order, debes asignar el resto de las propiedades del grupo **Absolute position**, para que veas correctamente ubicado al control en el canvas.
- Para que al seleccionar un valor del dynamic combobox de países, tenga efecto en la lista de parques, debes programar el evento ControlValueChanged de la variable del dynamic combobox con un comando Refresh y agregar una regla Parm con la variable del combo.

MÚLTIPLES LAYOUTS EN UN GRID

Se desea cambiar la información que muestra el panel ViewParks, dependiendo de ciertas condiciones. Antes de continuar, haz un Save As de este panel y guárdalo como ViewParksWithLayouts.

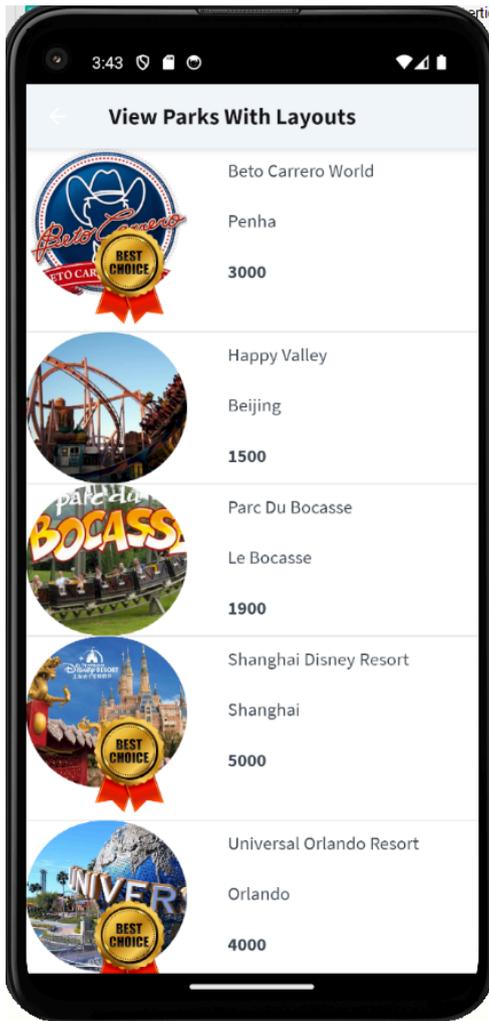
Por defecto se desea que muestre los parques de diversiones disponibles en la base de datos, mostrando la foto, el nombre, la ciudad y el promedio diario de visitantes del parque.

La pantalla con los datos por defecto deberá lucir parecida a la siguiente:



Para aquellos parques de diversiones que la cantidad de visitantes supera los 2000 diarios, automáticamente debe mostrarse el sello de “Mejor elección” superpuesto sobre la foto del parque. Además para esos parques el dato de la cantidad de visitantes debe mostrarse en color rojo y resaltado en negrita. Para el resto, no debe mostrarse el sello, ni el número de visitantes resaltado.

La pantalla con el sello de “Mejor elección” superpuesto debe verse similar a esta:

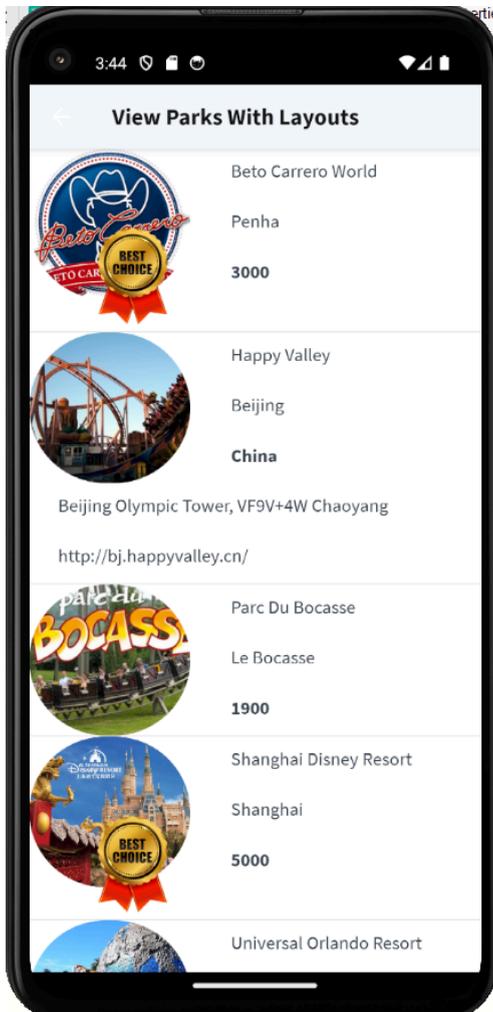


Sugerencias:

- Para poder asignar más de un layout a un panel, primero debes crear un layout nuevo, haciendo clic en la flecha de la esquina superior derecha de la tabla del grid y elegir Add New Item Layout.
- Para que el sello Best Choice se superponga a la foto de la atracción, puedes utilizar un **control canvas**. Para esto, deberás primero quitar todos los controles del grid, insertar el canvas al grid y la imagen del sello, luego insertar un control tabla e insertar los controles nuevamente.
- Una forma sencilla de hacer esto es primero que nada copiar la tabla del grid, luego borrar los controles de la tabla, insertar el canvas con la foto y luego pegar la tabla sobre el control canvas. Por último asignamos a la imagen de sello, un valor de Z-Order mayor al de la tabla que contiene los controles.
- No olvides asignar las propiedades del grupo **Absolute position** de la imagen y darle una altura de 100% a la tabla.
- Para asignar un layout dinámicamente, debes hacerlo en el evento Load del panel, chequeando la condición (en este caso que el número de visitantes sea mayor a 2000) y asignando la propiedad ItemLayout del grid con el nombre del layout correspondiente.

Además de lo anterior, al hacer tap sobre cualquier parte de la información del parque, ya no debe invocarse al panel que muestra los games, sino que debe ocultarse la cantidad de visitantes y mostrarse la dirección, el país y la página web.

La pantalla con más información de los parques debe verse como la siguiente:

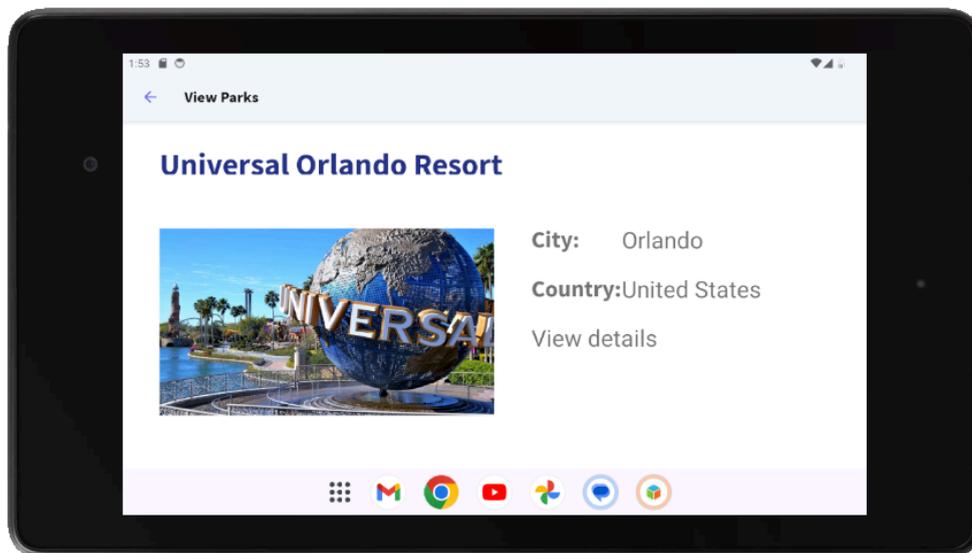


Recuerda:

- Hasta ahora estaba asignado el valor de la propiedad **Default Action** del grid a un evento que invocaba al panel ViewGames. Esto hacía que al hacer tap sobre uno de los parques, se abriera la pantalla que mostraba los juegos del parque seleccionado. Si quieres que al hacer tap se invoque a otro layout diferente del layout por defecto (Layout1), debes volver la propiedad Default Action a su valor por defecto y asignar la propiedad **Default selected item layout** con el nombre del layout que muestra el detalle del parque seleccionado.
- Para ejecutar al panel ViewParksWithLayout puedes agregarlo como action al MenuMain y asignarle una imagen como icono, o también puedes setear al panel como objeto main y ejecutarlo directamente.

MÚLTIPLES LAYOUTS PARA UN PANEL

Es necesario que la pantalla ViewParks reajuste su contenido cuando la aplicación se está usando en una Tablet de 7" en orientación horizontal, para que se vea parecida a la siguiente:



Sugerencias:

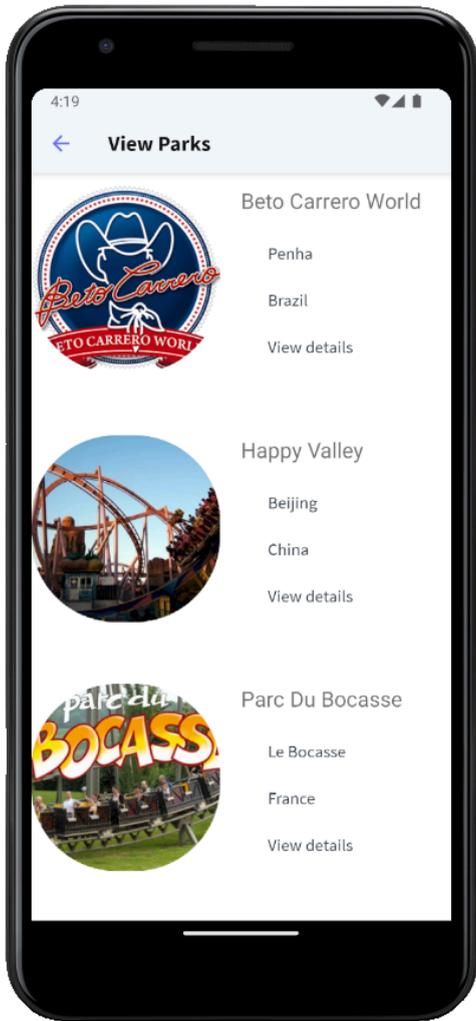
- Crea un nuevo device en el Android Studio para simular una Tablet de 7".
- No olvides crear un nuevo layout para el panel.
- Usa las propiedades **Columns Style**, **Rows Style** y **Col Span** de la/s tabla/s contenedora/s para definir el espacio que ocupan los elementos en las filas y columnas.
- Usa clases con las propiedades que usaste anteriormente para cambiar la apariencia del nombre de los parques y el tamaño de la información mostrada. Para cambiar el estilo de las etiquetas de los atributos, usa la gx-property `gx-label-class` y asigne una clase que cambie ese estilo.

Recuerda:

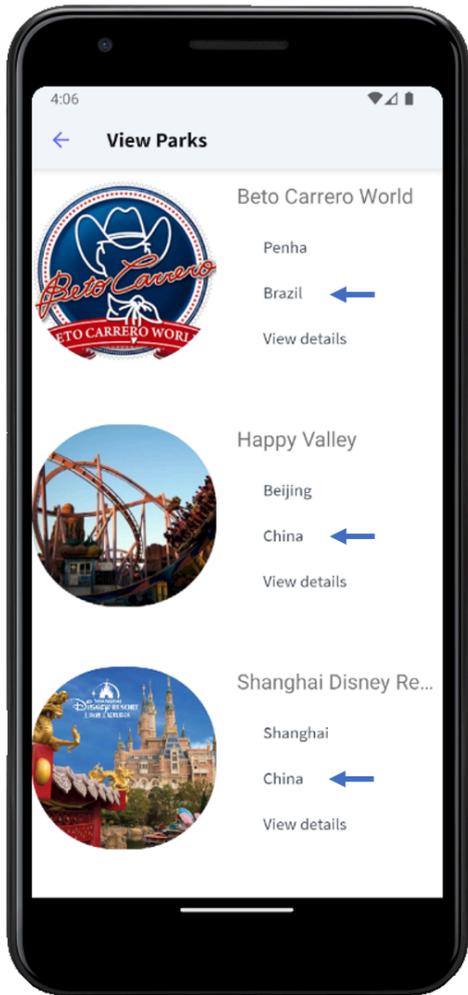
- Para visualizar un único registro por fila y desplazarte horizontalmente, puedes usar las propiedades **Scroll direction**, **Items Layout Mode** e **Items Per Column** o cambiar el tipo del Grid a **Horizontal**.

USO DE GRILLAS

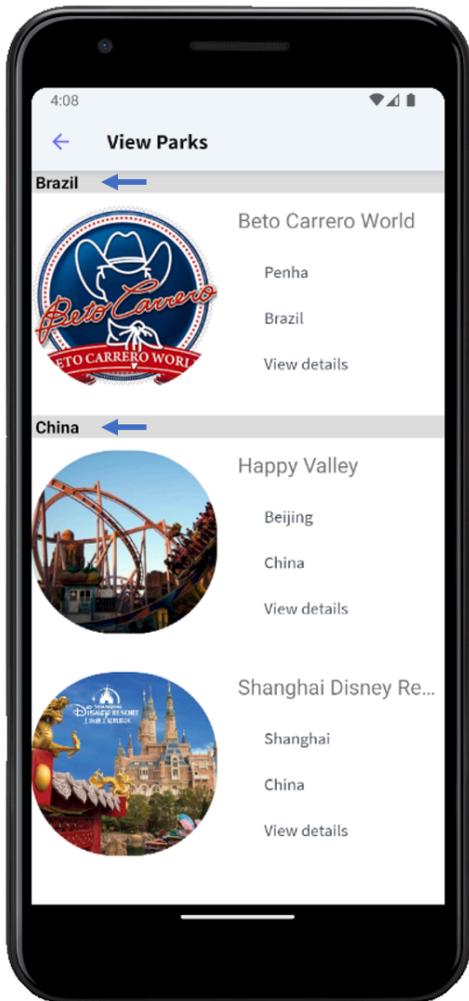
Modifica el layout por defecto del panel ViewParks para que el Grid tenga un desplazamiento horizontal y muestre como máximo 3 filas por página. La pantalla deberá verse parecida a la siguiente:



Ahora, los datos deberán salir ordenados de forma ascendente por nombre de país:



Ahora, es necesario que los parques de diversiones se vean agrupados por país:



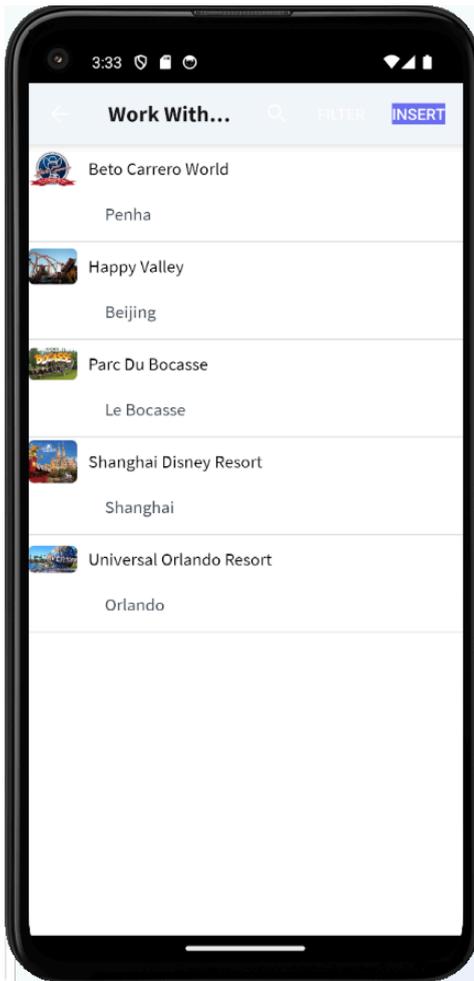
PATRÓN WORK WITH

Para poder realizar acciones de CRUD (Crear, leer, actualizar, eliminar) sobre los parques de atracciones, de una manera simple desde un dispositivo móvil, se requiere lo siguiente:

Aplicar el patrón Work With a la transacción AmusementPark

Agregarlo como ítem al objeto Menu MenuMain

Al acceder al listado de parques de atracciones a través del work with, se quiere que además del nombre y foto de cada parque, aparezca el nombre de la ciudad donde se encuentra, similar a como se muestra en la siguiente imagen:



Te sugerimos mejorar el diseño de esta pantalla.

También se quiere que al buscar, además de poder buscar por nombre del parque, en el mismo search también se pueda buscar por nombre de ciudad. Por lo que, cuando buscamos por nombre de parque nos traiga los parques que coincidan con ese nombre, y si buscamos por nombre de ciudad, nos traiga los parques que sean de esa ciudad.

Sugerencias:

- Para mostrar la ciudad junto al parque de atracciones, debemos hacerlo desde el nodo List del patrón WorkWith, agregando el atributo correspondiente
- Para buscar por más de un atributo, se deberá hacer la configuración en las propiedades del grid (que está en el nodo List) y configurar la propiedad "Search". Agregando el otro atributo por el cual deseamos buscar.

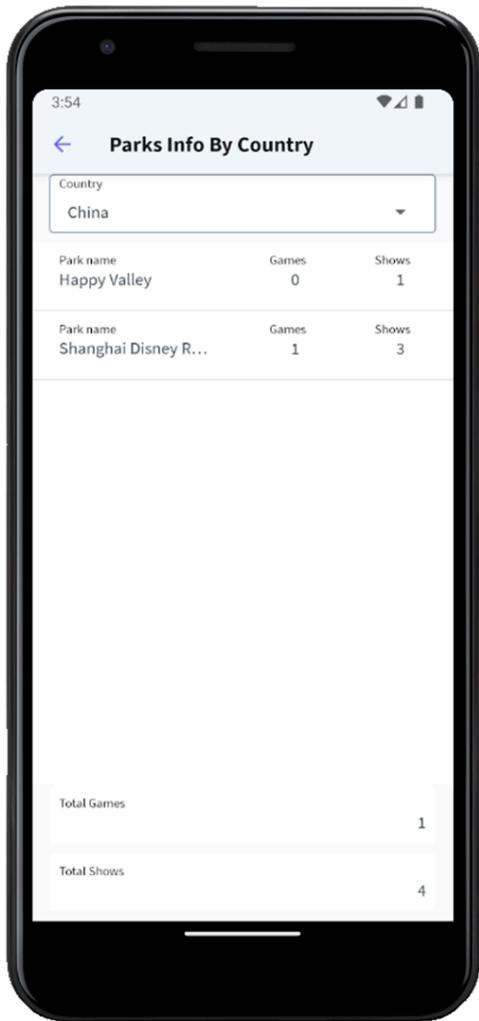
PROGRAMACIÓN DE EVENTOS EN UN PANEL

Es necesario desarrollar una pantalla que muestre para cada parque de diversiones su total de juegos y su total de shows, permitiendo filtrar por país. Además, debe mostrar debajo de esa lista los totales generales. La pantalla deberá verse parecida a la siguiente:

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:



Agregarlo como ítem al objeto Menu MenuMain.

Recuerda:

- Que los Panels se programan de manera diferente a los WebPanels, ya que la parte fija y la parte variable usan Data Providers diferentes para su carga.

Sugerencias:

- Usa los eventos Load y Refresh para cargar las variables del Grid y de la parte fija del Panel respectivamente.

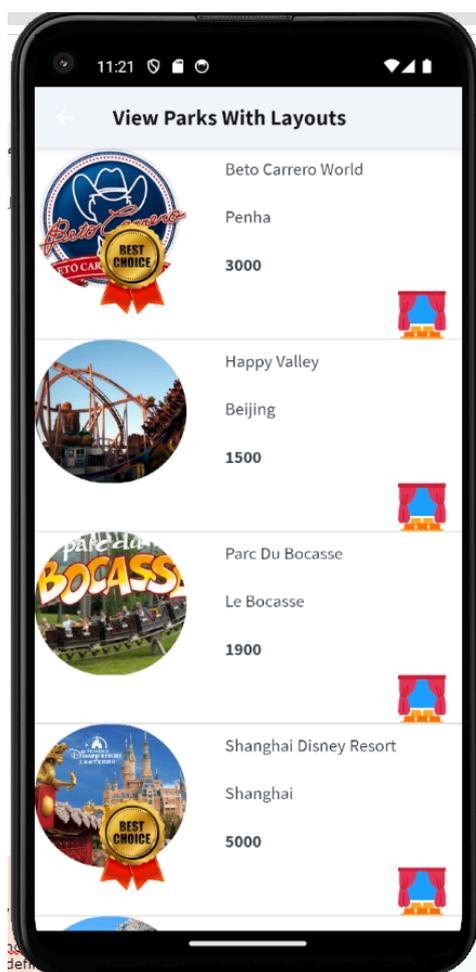
USO DE API's

Parte 1)

Se requiere invocar a una API que permita crear una cita en el calendario del dispositivo, de forma de agendar una función de un show previamente seleccionado.

Antes de continuar, es necesario poder ver la lista de shows de un parque de atracciones y luego visualizar el detalle del show que queremos visitar. Para esto, en el panel ViewParksWithLayouts se debe mostrar para cada parque, una imagen que simbolice los shows, para que al hacer clic sobre ella nos lleve a la lista de shows de ese parque.

El icono de shows en ViewParksWithLayouts puede verse de esta manera:

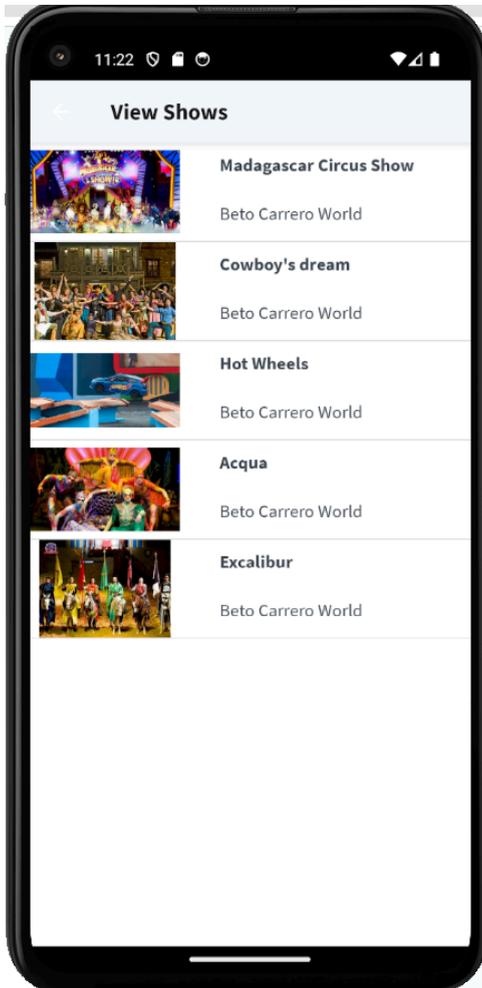


Sugerencia:

- Crea una variable &Shows del tipo Image y en evento Load del grid y asigna la imagen de los shows, con el método FromImage de la variable.
- Arrastra una tabla a la tabla del grid y dentro coloca la variable &Shows con la propiedad Label position en None. Esto deberías repetirlo para todos los layouts que tengas definidos que desees que accedan a los shows.

Al hacer clic sobre el icono de shows de un parque, debe abrirse el panel ViewShows con la lista de shows que tiene el parque, mostrando la foto y nombre de cada show, y el parque seleccionado.

Este panel debe lucir semejante al siguiente:



Luego, al hacer clic sobre un show, deberá abrirse el detalle del show invocando, para lo cual crearemos un panel llamado ShowMoreInfo, con la foto ampliada del show y los datos del nombre del show, la hora de comienzo y fin, la ciudad y el país.

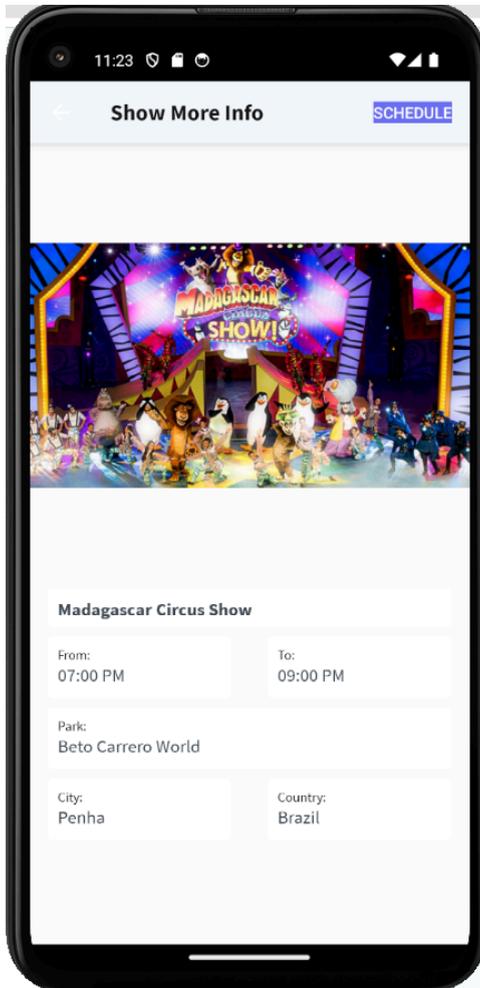
Para crear una cita en el calendario que nos recuerde asistir a la función de este show, agregamos un botón con el texto SCHEDULE, en cuyo evento invocamos a un panel que nos permita ingresar la fecha en que queremos ir al show.

Recuerda:

- En el panel ShowMoreInfo, dar el diseño adecuado a la tabla que contiene a los controles, ubicando a los mismos de forma de que se vean como la pantalla de ejemplo. Además no olvides asignar valores adecuados a las propiedades Column Style y Row Style de la tabla para visualizar el detalle del Show con la foto ampliada y los datos legibles.

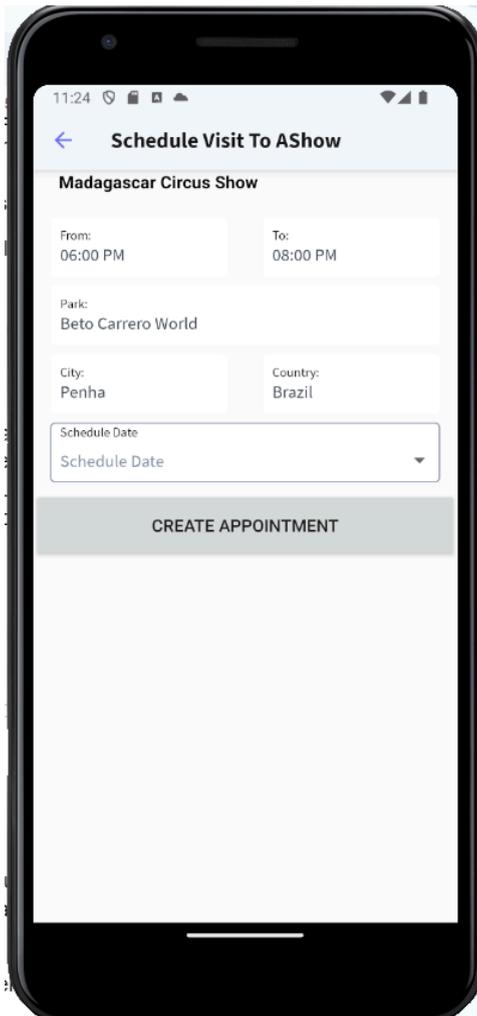
- El botón SCHEDULE debes insertarlo en el control Application Bar del pane, dando botón derecho sobre el mismo y seleccionando Insert button...

El panel ShowMoreInfo con la información del show y el botón SCHEDULE, debería verse como sigue:



El panel invocado por el botón SCHEDULE debemos crearlo con el nombre ScheduleVisitToAShow y nos mostrará los datos del show, incluidas las horas de comienzo y fin de la función, y deberá incluir la posibilidad de ingresar la fecha que elegimos para la función, ya que todos los días hay función a la misma hora.

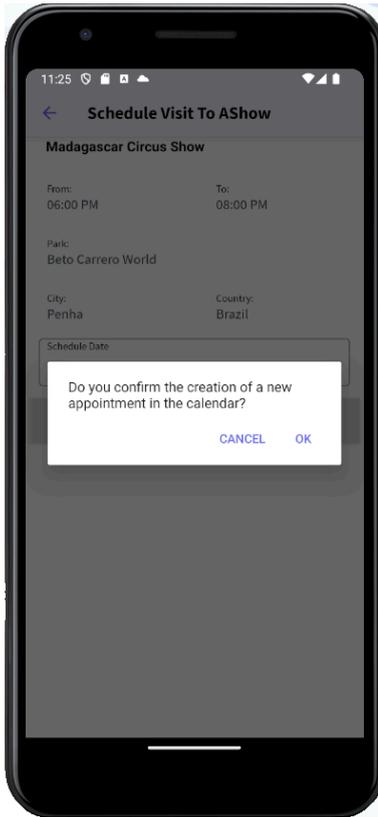
La pantalla del panel ScheduleVisitToAShow, deberá lucir similar a la siguiente:



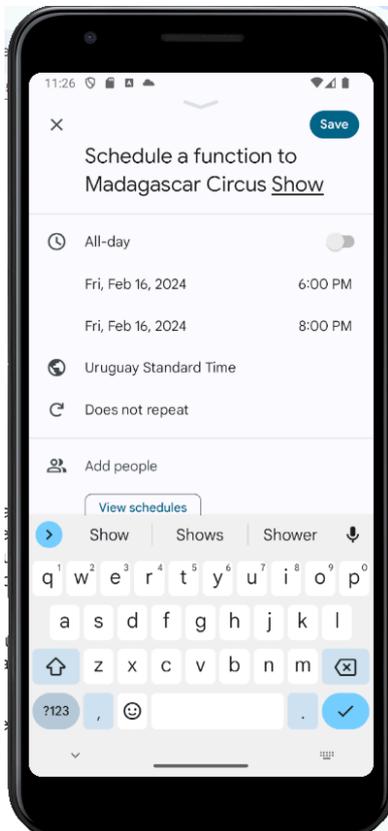
Luego con esos datos, se deberá invocar a una API que permita agregar una entrada en el calendario nativo del dispositivo. La invocación debe incluir un texto que indique que se hará una cita al show seleccionado (por ejemplo "Schedule a function to Madagascar Circus Show") pasando luego como parámetros los datos requeridos, que disponemos del show.

Antes de que se ejecute la invocación se deberá pedir al usuario que confirme la creación del nuevo evento en el calendario, dando la posibilidad de confirmar o declinar la creación de la cita del calendario.

El popup con el diálogo para confirmar la acción de crear la cita en el calendario deberá verse como se muestra:



y al presionar el botón Confirmar, se deberá ver la cita del calendario a guardar.



Recuerda:

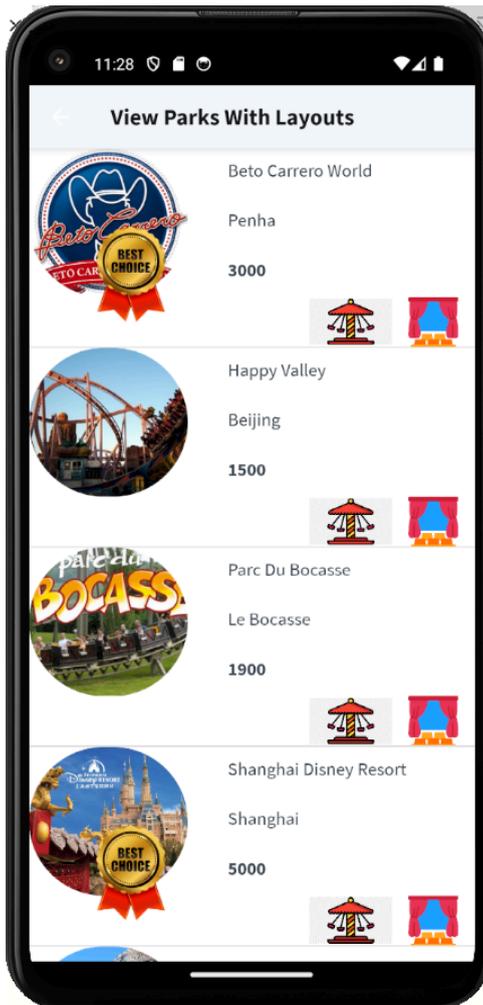
- Para hacer que aparezca un diálogo de confirmación, debes invocar al método Confirm de la API Interop, del nodo References, módulo GeneXus.SD, pasándole como parámetro el mensaje correspondiente.
- Para invocar una API que agregue una cita en el calendario del dispositivo, debes utilizar el método Schedule de la API Calendar, ubicada en el módulo GeneXus.SD, pasando por parámetro el texto de la cita, la fecha de comienzo, fecha de fin, hora de comienzo, hora de fin y un texto con el lugar de la cita.

Parte 2)

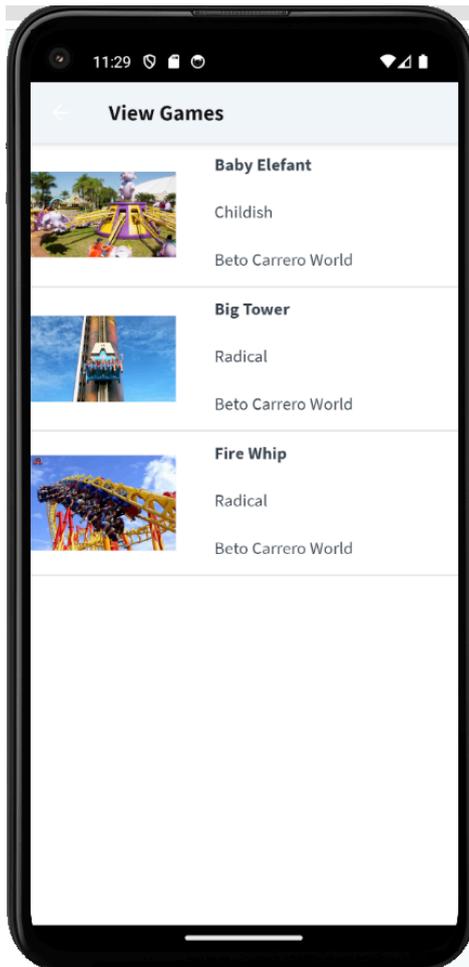
Como parte de la promoción de los parques de diversiones, se desea hacer propaganda de un juego de un parque, mediante un mensaje en la plataforma X (Twitter), mostrando en el mensaje un texto de un máximo de 280 caracteres. El texto del mensaje debe incluir una invitación a visitar el juego, que mencionará el nombre del juego, el parque al que pertenece, la ciudad y el país, por ejemplo: “Enjoy the game Hulk at Shanghai Disney Resort amusement park located in Shanghai, China”.

Para implementar esto, deberá agregarse al panel ViewParksWithLayouts un icono para ver los juegos de ese parque, en forma análoga a cuando agregamos el icono para ver los shows.

La pantalla ViewParksWithLayouts deberá verse como sigue:

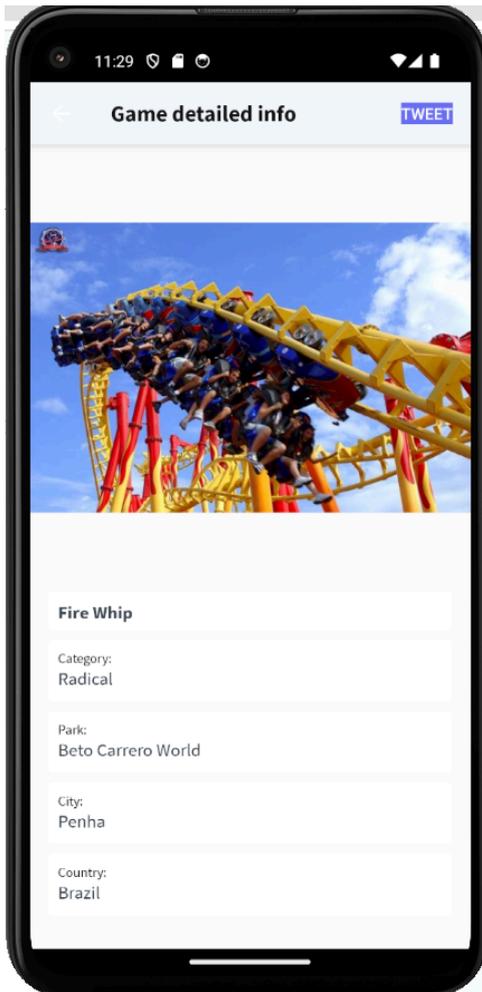


Al presionar el icono de juegos, se deberá invocar al panel ViewGames, que debería verse como se muestra a continuación (agregale el diseño adecuado para eso):



Al hacer clic sobre un juego, deberá abrirse un panel que llamaremos GameMoreInfo con el detalle del juego, que mostrará la imagen del juego ampliada y debajo de ella el nombre del juego, su categoría, el parque al que pertenece, la ciudad y el país. También se agregará un botón TWEET en la Application Bar, que ejecutará la acción de publicar el twit.

La pantalla del panel GameMoreInfo debe parecerse a ésta:

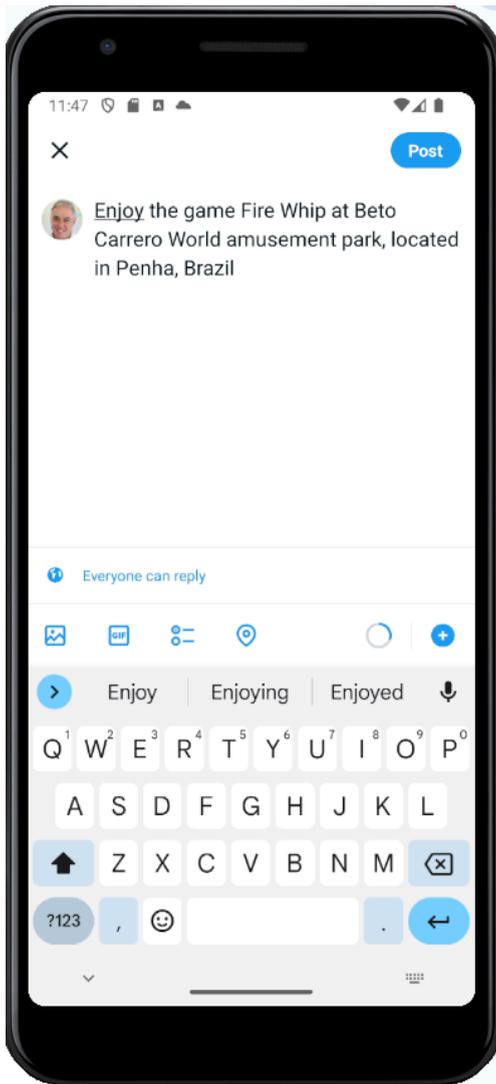


Sugerencias:

- En el panel GameMoreInfo, dar el diseño adecuado a la tabla que contiene a los controles, ubicando a los mismos de forma de que se vean como la pantalla de ejemplo. Además no olvides asignar valores adecuados a las propiedades Column Style y Row Style de la tabla para visualizar el detalle del Game con la foto ampliada y los datos legibles.
- El botón TWEET debes insertarlo en el control Application Bar del panel, dando botón derecho sobre el mismo y seleccionando Insert button...

Al presionar el botón TWEET, debe invocarse a la API que publica el twit, pasándole como parámetro el texto antes mencionado.

Si se tiene la aplicación X instalada en el dispositivo ésta se abrirá mostrando el mensaje a publicar, de lo contrario se abrirá el browser del dispositivo situado en la página de X, mostrándose el twit a ser publicado.



Recuerda:

- Para invocar una API que publique un Tweet, debes utilizar la API Twitter, ubicada en el módulo GeneXus.Social, pasando por parámetro el texto del tuit.
- Si tienes la aplicación X instalada en el emulador se abrirá la aplicación, de lo contrario se abrirá la página del navegador ubicada en el sitio de X. Estas acciones pueden requerir que te loguees en la plataforma para continuar.

PANEL CON FUNCIONAMIENTO OFFLINE

Interesa que toda la aplicación pueda ser también utilizada de manera offline.

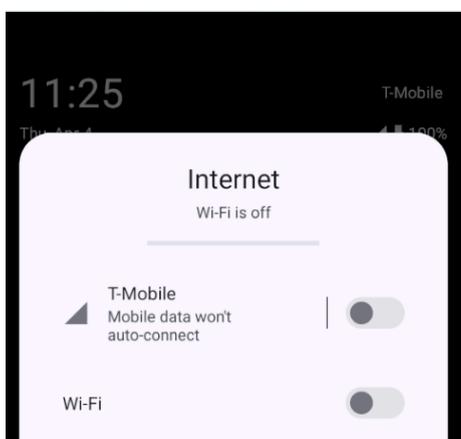
Por lo cual, cuando se acceda con conexión a internet, la aplicación se conectará con el server y estará sincronizada con la base de datos. Pero si se accede sin conexión a

Internet, queremos que igualmente se pueda navegar sin problemas por las diferentes pantallas.

Sugerencias:

- Para cumplir con el requisito, deberás configurar la propiedad Connectivity Support, del objeto MenuMain, en el valor Offline.

Luego de realizar el procedimiento correspondiente, para probar que efectivamente tenemos la aplicación funcionando de manera offline, desconecta la aplicación de internet (apaga Wifi y datos móviles)



Ingresa al WorkWith AmusementPark, y agrega un parque de atracciones nuevo.

Para corroborar que este cambio haya sido realizado solamente en la base de datos del dispositivo, corrobora en la interfaz web, accediendo a WWAmusementPark mediante el Launchpad (View -> Other Tool Windows -> Launchpad), que el registro que acabas de ingresar no aparezca aún.

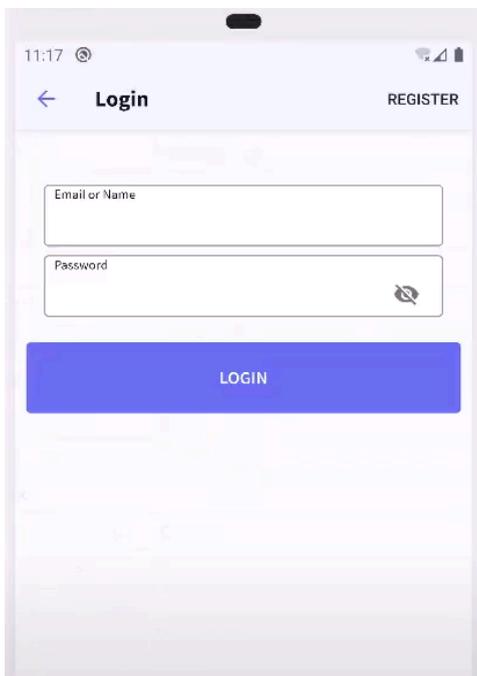
Luego vuelve a conectar el dispositivo móvil a internet, cierra y abre la aplicación para que se realice la sincronización de los datos con el server, y luego ingresa nuevamente a la interfaz web de WWAmusementPark a través del Launchpad. Ahora sí debería aparecer el nuevo parque de atracciones que ingresaste estando offline.

AGREGANDO SEGURIDAD A LA APLICACIÓN

Aplicar GAM a la aplicación, para que al ejecutar cualquier objeto main, nos pida un logueo de autenticación. En este caso lo probaremos accediendo al objeto menu MenuMain.

Recuerda:

- Para aplicar GAM, deberás poner la propiedad Enable Integrated Security en True, en el nodo de la versión de la KB, en el KB explorar.
- Para poder hacer el login en la pantalla generada por el GAM, es importante que el dispositivo móvil tenga conexión a Internet, aunque la app siga siendo offline.



Para poder implementar lo que viene a continuación, debemos volver a tener una aplicación Online. Por lo cual deberás volver a configurar la aplicación en ese modo.

Sugerencias:

- Para cumplir con el requisito, deberás configurar la propiedad Connectivity Support, del objeto MenuMain, en el valor Online.

Se necesita crear un usuario (con el nombre y contraseña que elijas), desde el backoffice web del GAM, que al loguearse sólo tenga permiso para acceder al objeto WorkWithAmusementPark.

Vuelve a ejecutar la aplicación y logueate con este nuevo usuario para comprobar que efectivamente sólo puedes acceder al objeto WorkWithAmusementPark, y no al resto de los objetos agregados al MenuMain

Sugerencias:

- Para crear un nuevo usuario de logueo, deberás ir dentro del KB explorer a la carpeta GAM_Examples, y luego ejecutar (run) el objeto GAMHome
 - En el nodo de la base de conocimiento, deberás cambiar la propiedad Integrated Security al valor Authorization, para que además de autenticar, controle los permisos.
-



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.