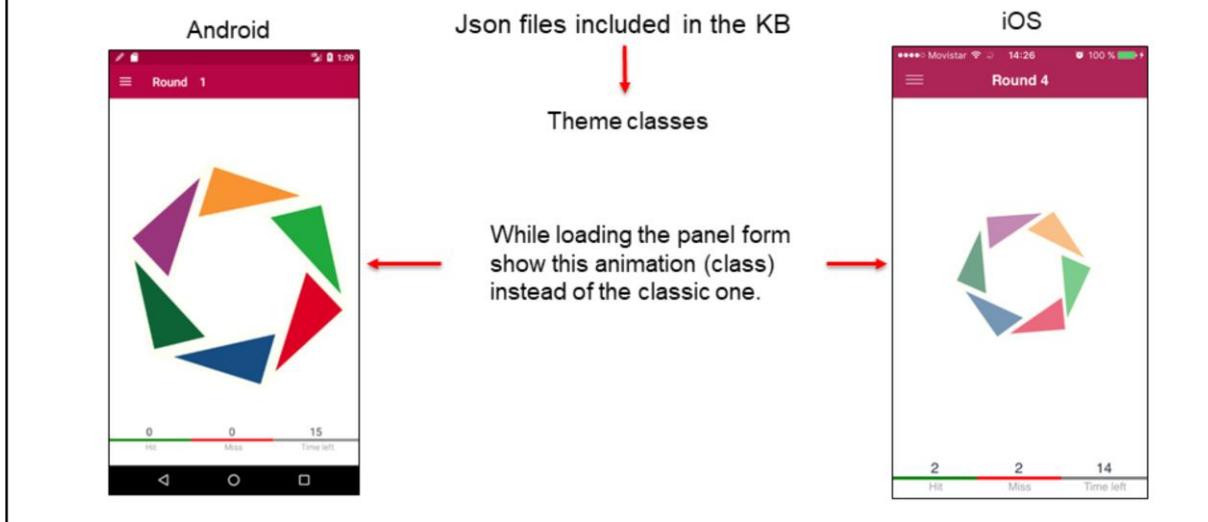


# Smart Devices

*GeneXus™ 15*

# Animations

## Animations



A partir del upgrade 8 pueden integrarse animaciones, específicamente animaciones Lottie (una library creada por Airbnb, que provee una api Json para integrar las animaciones en aplicaciones mobile). Con esos archivos json podemos incluir animaciones en la KB.

Hay muchos repositorios de animaciones de los que pueden descargarse en formato Json.

Habr a que incluirlos en la KB y crear clases para cada uno de ellos.

En el ejemplo, que vimos en la aplicaci n de las canciones, estamos utilizando una animaci n para personalizar el circulito que por defecto ya aparece cada vez que se est  haciendo un Loading de una p gina o grid.  C mo lo implementamos?

## Animations

The screenshot displays the GeneXus IDE interface for configuring a game panel. On the left, a mobile app preview shows a game screen with a circular logo and a progress bar. The central component inspector shows the 'PanelGameQuiz' component with a 'Table: MainTable' containing a 'Form' property. The class explorer on the right shows the project structure, with 'FormGame' selected. Two tables provide detailed configuration:

Class: FormGame	
Name	FormGame
Description	
Enter Effect	Fade
Exit Effect	Fade
Call Type	Replace
Target Name	
Content Size Change	Default
Loading Animation Class	AnimationLoader

Class: AnimationLoader	
Name	AnimationLoader
Description	
Type	Lottie
File	anim_loader

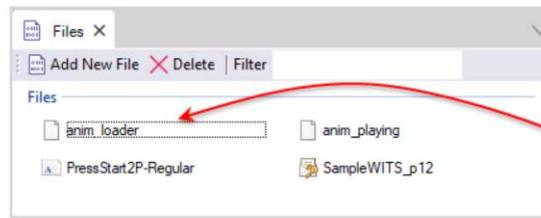
Acá vemos el form del panel que carga la pantalla con la ronda actual del juego. Si observamos las propiedades de la MainTable del Layout, veremos que a la Form Class le hemos asociado una clase FormGame que creamos en el theme, y a la que a su vez le hemos configurado la propiedad "Loading Animation Class" con el valor de una Animation class que también creamos.

A esa animación, a la que llamamos AnimationLoader, la definimos del tipo Lottie, y tuvimos que indicarle el nombre del archivo json de la animación, que previamente hubimos de insertar en la KB.

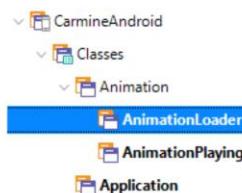
A continuación vemos los pasos de forma más ordenada.

## Animations: how to incorporate to the KB

1. Add the Lottie Json files as File objects in the KB



2. Create Animation classes in the Theme



Class: AnimationLoader

Name	<b>AnimationLoader</b>
Description	
Type	<b>Lottie</b>
File	<b>anim_loader</b>

¿Cómo incorporar entonces una animación a la KB?

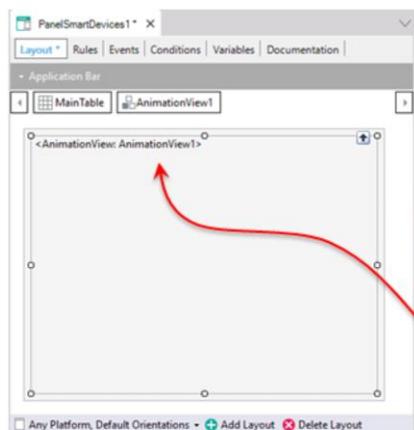
Primero agregamos los archivos json como objetos files, en la KB.

Luego, creamos animation classes en el theme. Por cada animation class apuntamos al archivo que subimos en el paso anterior.

## Animations



3. Use it in some theme class property where an animation is expected (eg: **Loading animation class** of Form, Grid and Matrix theme classes) or use it through the **Animation View control**:



### AnimationView methods:

- SetAnimation( AnimationClass, loop )
- SetProgress( progress )
- Play()
- Pause()

eg.

```
Event ClientStart
composite
  AnimationView1.SetAnimation(ThemeClass:AnimationPlaying, true)
  AnimationView1.Play()
endcomposite
endevent
```

Y luego la usamos en todo lugar, en toda clase del theme donde una animación es esperada. Por ejemplo en la propiedad Loading animation class tanto de clases de form, grid o matrix. Pero también podemos usarla con el nuevo control Animation View.

Este control se inserta en el layout y tiene estos cuatro métodos.

En el primero seteamos cuál será la clase de tipo Animation que será cargada en el control y si queremos que la reproduzca en loop o no. SetProgress permite indicar un número entre 0 y 100 para que desde allí arranque la animación. Las demás son obvias.

Aquí vemos un ejemplo. En el evento ClientStart que se ejecuta al principio de este panel, seteamos la animación (va a ser la que esté configurada en esta clase del tema) en loop... Y luego hacemos Play.

## SD Components and Global Events