

GeneXus[™]
The power of doing.

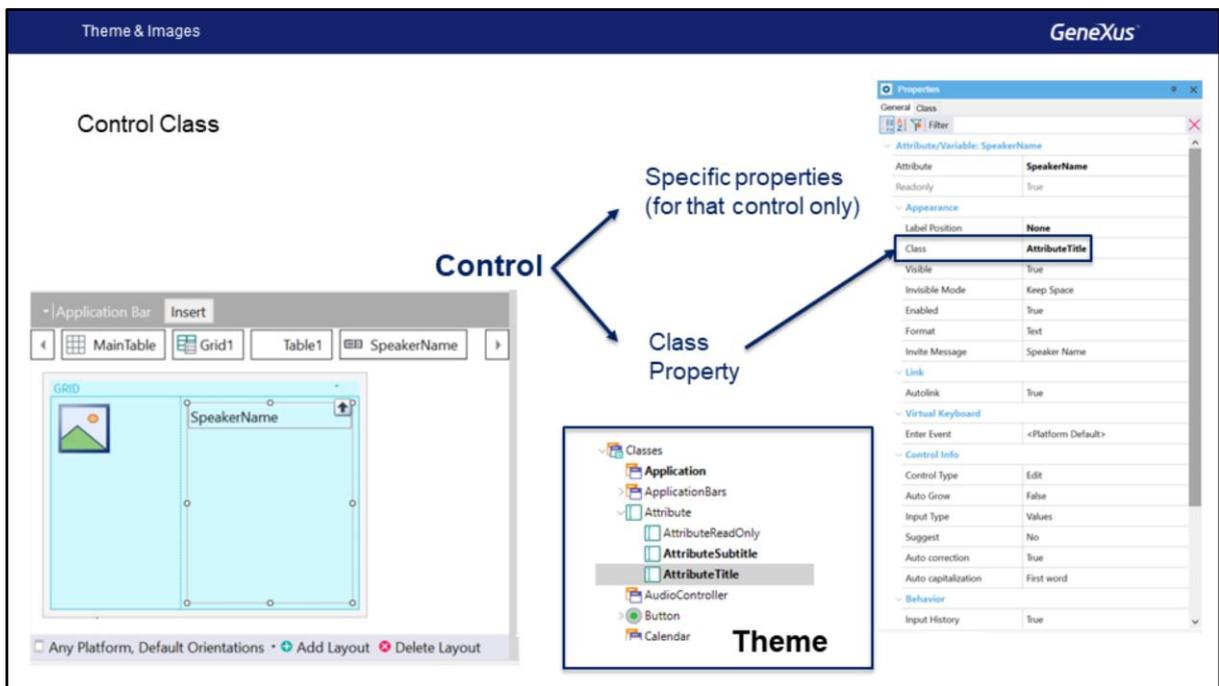
Theme & Images

User Interface & User Experience

GeneXus 16

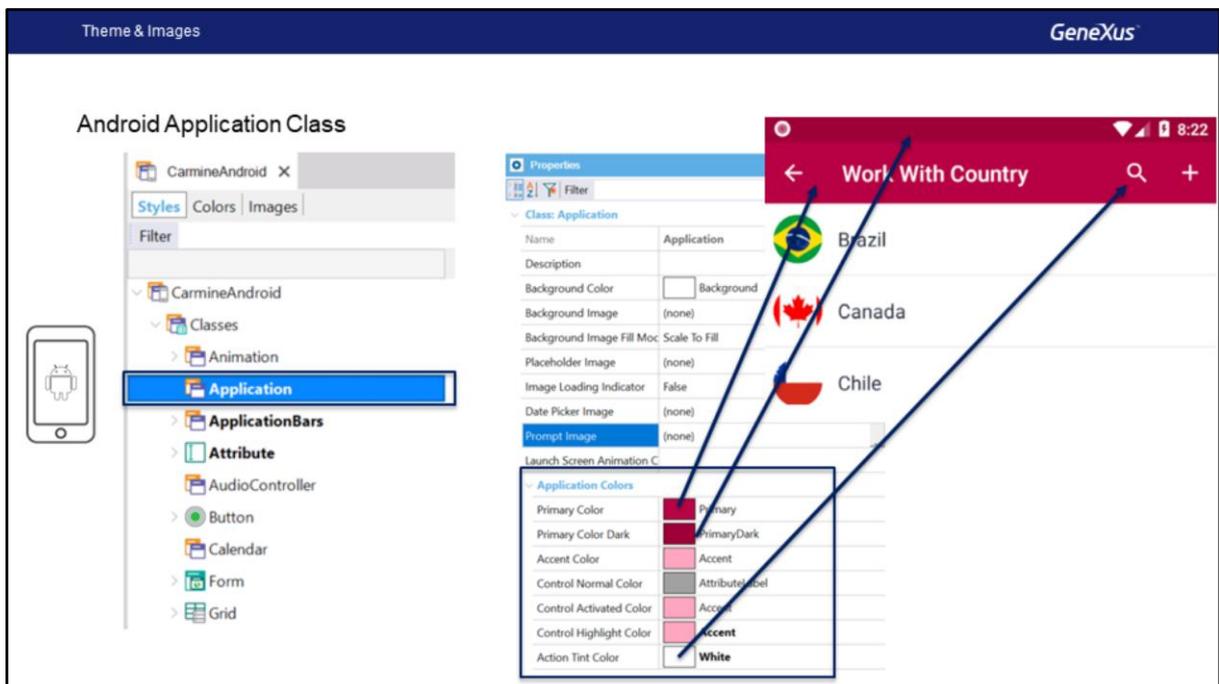
- 1 Theme Object
- 2 Color Pallete
- 3 Platforms Setup
- 4 Images Setup

Neste vídeo começaremos estudando o Objeto Theme e algumas de suas principais características, também veremos o uso da paleta de cores. Em seguida, continuaremos com a configuração dos temas nas plataformas Android e iOS e, finalmente, veremos a configuração das imagens.



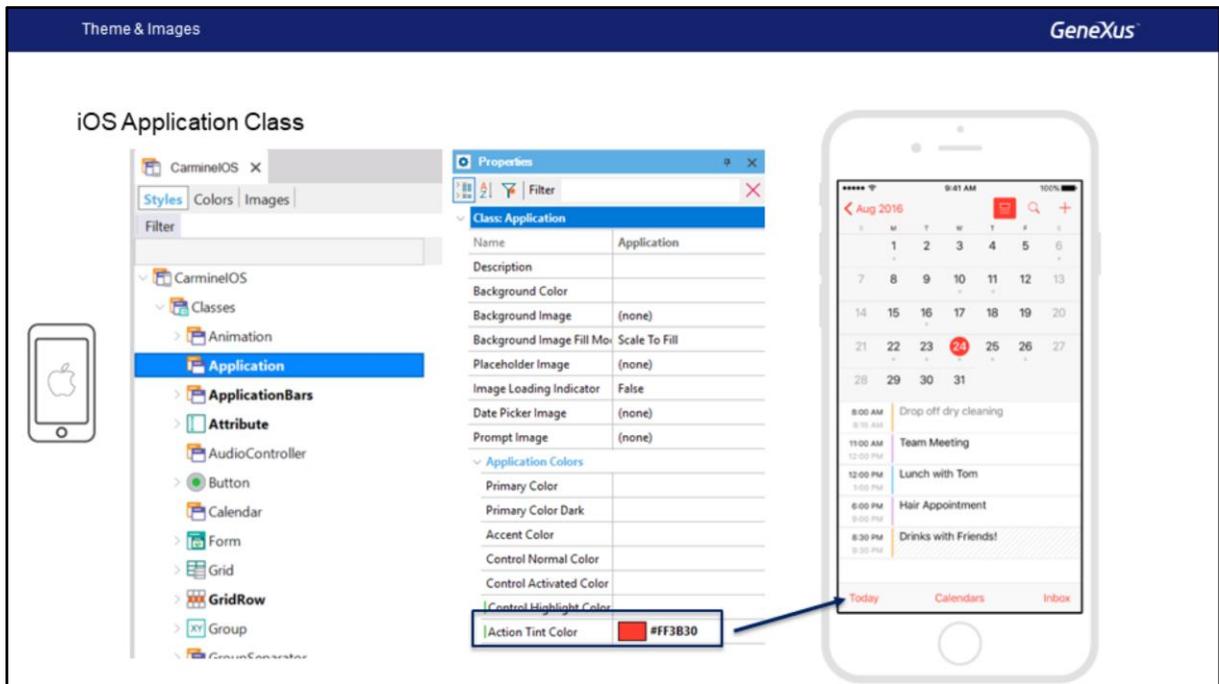
Como sabemos, em toda tela temos controles (atributos ou variáveis) e outros controles como imagens, botões, grids, etc. Cada controle tem suas propriedades personalizáveis, que indicam, por exemplo, propriedades do título, visibilidade, se está ativado. Mas, além disso, cada controle tem uma classe associada. É na classe onde são especificadas outras propriedades como cores, a fonte a ser usada, por exemplo. As classes permitem definir alguns aspectos apenas uma vez, para poder aplicá-los a muitos controles (do mesmo tipo) ao mesmo tempo.

Como a estética e o comportamento geral da app devem ser uniformes e de acordo com os guias de cada plataforma, é no objeto Theme, que contém todas as classes predefinidas e aquelas que adicionamos, nas quais faremos boa parte do trabalho de desenho e comportamento para alcançar a UI e UX que os designers determinaram.



Se formos ao theme default para Android, vemos que na classe **Application** existe um conjunto de propriedades agrupadas sob o nome Application Colors, que serão as que permitirão definir as cores de acordo com os critérios de Material Design que vimos.

Por padrão, essas propriedades vem pré-configuradas e são aquelas que vimos quando executamos a aplicação pela primeira vez, vemos que a Primary color se utiliza para a Application Bar, a cor Primary Color Dark se utiliza na Status Bar e, por exemplo, as ações utilizam a cor Action Tint Color.



Para iOS, a única propriedade do grupo Application Colors que tem efeito é Action Tint Color. Esta será a cor que as ações assumirão.

Theme & Images GeneXus

Custom Classes

TableTrackBullet

- TableTrackBullet001
- TableTrackBullet002
- TableTrackBullet003
- TableTrackBullet004
- TableTrackBullet005
- TableTrackBullet006
- TableTrackBullet007
- TableTrackBullet008
- TableTrackBullet009
- TableTrackBullet010

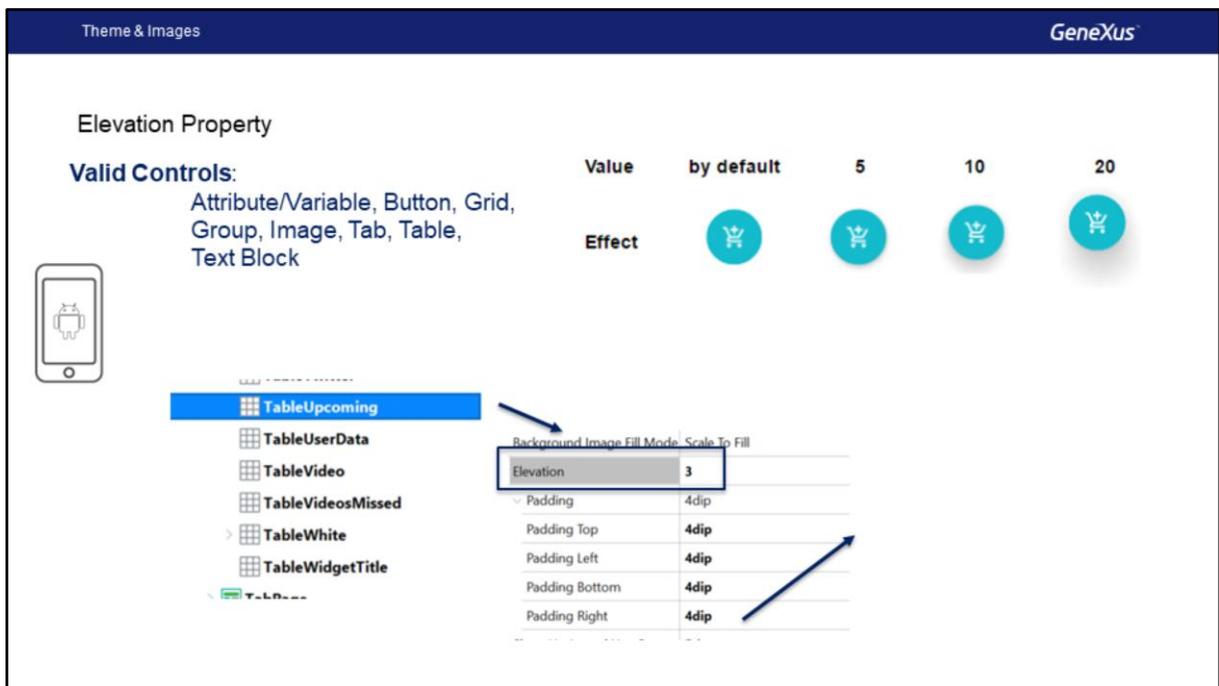
Background Color	TrackColor001
Highlighted Background Color	TrackColor001
Background Color	TrackColor003
Highlighted Background Color	TrackColor003

- AttributeTrackName
- AttributeTrackSp
- AttributeTrackTitle
- AttributeUserName
- AttributeVideoTitle
- CFDate
- CFDateResponse
- CFResponseMessage

Margin Left	30dip
Margin Bottom	
Margin Right	30dip
Background Color	
Highlighted Background Color	
Background Image	(none)
Highlighted Background Image	(none)
Forecolor	White
Highlighted Forecolor	
Font	.. 14dip _ San Francisco _
Font Weight	
Font Style	
Font Size	14dip
TextDecoration	
Font Family	San Francisco
Font Color	

Como desenvolvedores, com base nas informações que o designer gráfico nos enviará, modificaremos ou criaremos classes e subclasses nos themes, que serão então utilizadas pelos controles que inserimos nos layouts.

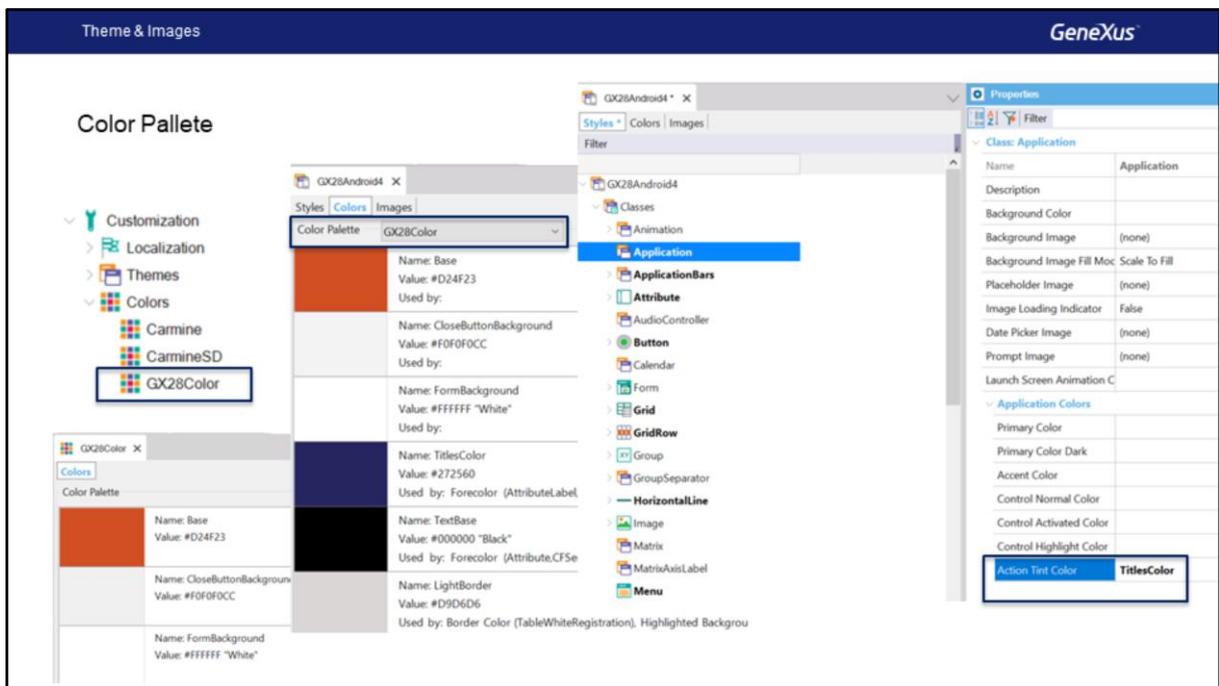
Aqui, estamos vendo um exemplo que posteriormente refletiremos em nossa app: a classe TableTrackBullet e suas subclasses, cada uma com uma cor característica de cada Track, ou a classe AttributeTrackTitle para mostrar os nomes das Tracks.



Ou, por exemplo, podemos especificar que controles da app Android (de alguns dos tipos indicados) podem ter um sombreamento, de acordo com o valor da propriedade Elevation. Desta forma, estarão sendo seguidos os guias do Material Design.

Aqui vemos como seria visualizada uma imagem com um valor default da propriedade Elevation (ou seja, sem especificar), com um valor de 5, de 10 e de 20.

Particularmente na app Event Day, vamos utilizá-lo, por exemplo, na classe TableUpcoming, para mostrar certos conteúdos, como a Lista de Palestrantes em Destaque que vemos à direita.



Os designers geralmente definem uma paleta de cores e então aplicam essa paleta ao desenho geral. Podemos criar um objeto de tipo paleta, definir as cores nele, dando-lhes um nome e, em seguida, ao Theme associar essa paleta em sua guia Colors, para que possa ser usada posteriormente nas classes..

Isto não é obrigatório, mas é uma boa prática, para não ter que inserir sempre as cores nas classes por seu valor, mas fazê-lo diretamente por seu nome, poderíamos também criar várias paletas e redefinir as cores em cada uma, tornando independente o Theme de cada cor em particular.

Theme & Images GeneXus

Multiplatform

Platform: Any Android

Name	Any Android
OS	Android
Version	
Device Kind	All
Size	All
Theme	CarmineAndroid
Additional Themes	(none)
Navigation Style	Default
Default Layout Orientation	Any
Bounds Name	
Minimum Shortest Bound	0
Maximum Shortest Bound	0

Platform: Any iOS

Name	Any iOS
OS	iOS
Version	
Device Kind	All
Size	All
Theme	CarminelIOS
Additional Themes	(none)
Navigation Style	Default
Default Layout Orientation	Any
Bounds Name	
Minimum Shortest Bound	0
Maximum Shortest Bound	0
Minimum Longest Bound	0
Maximum Longest Bound	0

Customization

- Localization
- Themes
 - Carmine
 - CarmineSD
 - CarmineIOS
 - CarmineAndroid
 - SD Platforms
 - Any Platform
 - Any Android
 - Android Phone
 - Android Tablet 7"
 - Android Tablet 10"
 - Any iOS
 - iPad
 - iPhone
 - iPhone 3.5"
 - iPhone 4"
 - iPhone 4.7"

Dizíamos que nossas apps para Smart Devices serão multiplataforma. Ou seja, basicamente criamos a mesma app e faremos o mínimo trabalho necessário para personalizá-la de forma que seja executada em dispositivos Android e em dispositivos iOS, seguindo seus guias de desenho e comportamento.

Contaremos em cada Base de Conhecimento com dois themes já criados, CarmineIOS e CarmineAndroid, e por padrão, esses Themes estão associados à plataforma Android, e outro à plataforma iOS, e será o mesmo para toda a plataforma, independentemente se for um Telefone ou Tablet, e de seu tamanho e resolução, mas poderemos criar outros themes para fazer variar o desenho e o comportamento geral, dependendo dessas dimensões.

Demo: Configuration of Themes and Platforms

Vamos ver tudo isto no GeneXus.

Primeiro veremos que já temos na KB os temas criados, GX28Android4 e GX28IOS, também temos a paleta de cores GX28Color.

Vamos abrir o tema GX28Android4, veremos na aba Colors, nesta propriedade Color Palette podemos alterar e atribuir a paleta de cores a um tema específico.

Como recomendação, usar nomes que não façam referência à cor, por exemplo, FormBackground.

Bem, vamos agora na classe Application à Action Tint Color e vamos fazer referência a um nome da paleta, TitlesColor, também podemos inserir e selecionar uma cor da paleta. Bem, vamos gravar.

Vamos agora para a plataforma Any Android e vamos associar o tema, selecionamos Gx28Android4, neste caso, para todas as plataformas Android, será o mesmo. Fazemos o mesmo em Any IOS, selecionamos GX28 IOS.

Bom. Antes de executar, vamos alterar no objeto EventDay, no menu, Menu for Smart Devices, a propriedade Android Base Style, vamos usar Light em vez de Light with Dark Action Bar. Isto nos dará o efeito que estamos buscando na aplicação.

Vamos executar a aplicação, bem, esperamos um momento.

Bem. Já temos a aplicação instalada no emulador, vemos que mudou a cor Vermelha, esse tom de vermelho na Application Bar, veremos por exemplo, temos as ações que estão assumindo a Action Tint Color.

Veremos, por exemplo, o Uruguai, aqui vemos, por exemplo, a propriedade Tabs Strip, nesse tom de verde, não tem uma cor específica porque está tomando o valor padrão da plataforma.

Com isto, termina a demonstração, continuaremos vendo os tópicos seguintes.

Setting up an Image

The screenshot displays the 'Images' window in the GeneXus IDE, showing a list of image assets. The 'Properties' window is also open, showing the details for the selected image asset.

Image	Theme	Language	Density
ic_launcher (3).png Size:48x48 px (48x48 dip)	Any	Any	100% (mdpi Android, 1x iOS, 1x Web)
ic_launcher (5).png Size:72x72 px (48x48 dip)	Any	Any	150% (hdpi Android, 1.5x Web)
ic_launcher (2).png Size:96x96 px (48x48 dip)	Any	Any	200% (xhdpi Android, 2x iOS, 2x Web)
ic_launcher (1).png (Default) Size:144x144 px (48x48 dip)	Any	Any	300% (xxhdpi Android, 3x iOS, 3x Web)
ic_launcher.png Size:192x192 px (48x48 dip)	Any	Any	400% (xxxhdpi Android, 4x Web)
ic_launcher (4).png Size:36x36 px (48x48 dip)	Any	Any	75% (ldpi Android, .75x Web)

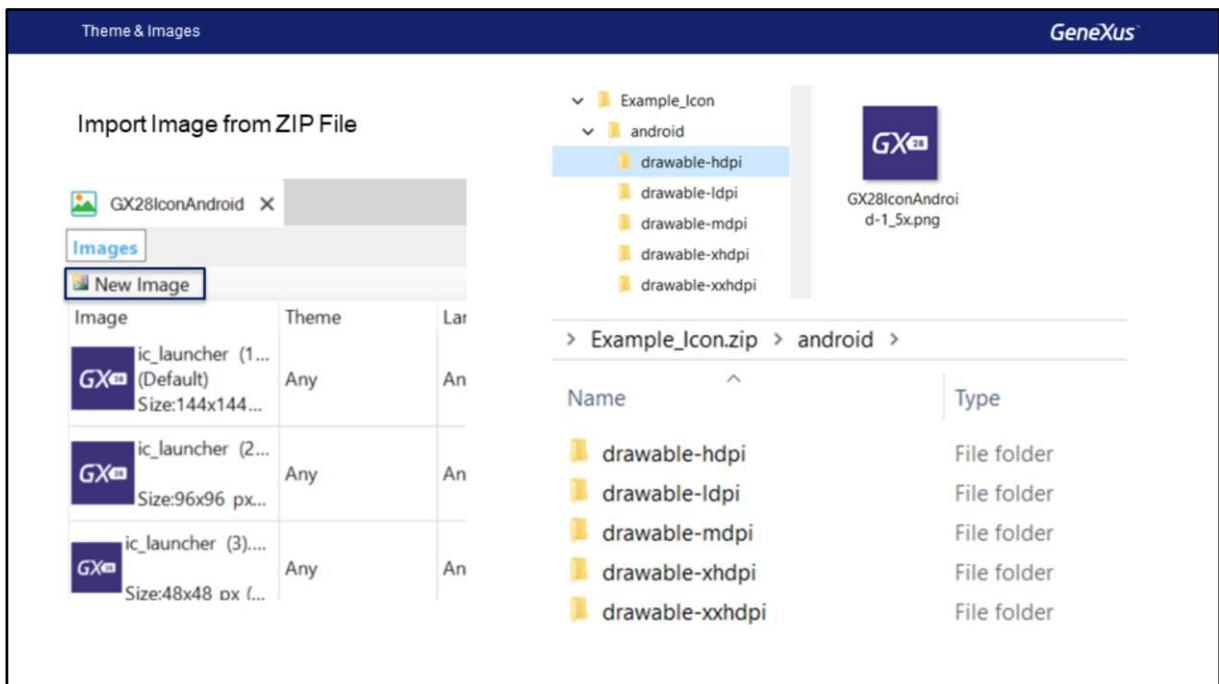
Properties for Image: ic_launcher (3).png

Name	ic_launcher (3).png
Description	ic_launcher (3).png
Is External	False
Theme	Any
Language	Any
Density	100% (mdpi Android, 1x iOS, 1x Web)
Rendering Mode	Automatic
Source	C:\Fabian\GX28\Assets\Icono\ic_launcher (

Vamos ver alguns aspectos sobre as imagens.

Aqui temos o exemplo do ícone da aplicação. Observemos que criamos um único objeto do tipo imagem na KB, que chamamos de GX28IconAndroid e dentro inserimos cada arquivo com a imagem que o designer nos enviou, podendo variar as mesmas por Theme, densidade e idioma.

Neste caso, no objeto main da aplicação, simplesmente configuramos este objeto imagem e dependendo da plataforma e da resolução do dispositivo, a imagem correspondente será usada.



Para adicionar as diferentes versões da mesma imagem, temos duas opções.

A primeira consiste em criar uma nova imagem e depois adicionar manualmente com a opção New Image as outras versões e configurar para cada uma as densidades correspondentes.

A outra é, Importar as imagens de um arquivo ZIP, neste caso, as imagens devem estar organizadas dentro do arquivo ZIP, como mostrado na imagem Example_Icon.ZIP, deve ter uma pasta Android e então para cada densidade uma pasta com a imagem correspondente.

Em seguida, Criamos uma nova imagem usando a opção Create Image from File, selecionamos o arquivo ZIP, no nosso caso Example_Icon.ZIP e pressionamos o botão Finalizar.

O resultado é aquele que vemos nesta imagem à direita, onde todas as densidades são configuradas automaticamente ao importar o arquivo ZIP. Leve em conta esta característica ao solicitar as imagens ao designer, que já pode fornecê-las organizadas de forma que possam facilmente importá-las para a base de conhecimento, economizando muito tempo.

Theme & Images GeneXus

Setup Main Object Images

 **Android**

Android Version Code	1.0
Android Version Name	1.0
Update URL	
Android Package Name	com.artech.eventday.eventday
Android Application Icon	GX28IconAndroid
Android Notification Icon	(none)
Android Portrait Launch Image	GX28LaunchAndroid
Android Landscape Launch Image	GX28LaunchAndroidLandscape

 **Apple**

Apple Version Code	1.0
Apple Version Name	1.0
App Store URL	
Apple Bundle Identifier	com.artech.EventDay
Development Team ID	
Devices	iPhone & iPad
Watch Main	(none)
Apple Application Icon	GX28Icon
Apple Launch Image	GX28Launch

Vamos repassar isto que já vimos.

Lembre-se que é necessário ter as imagens de **launch** (para orientação Portrait e Landscape) e o **ícone da aplicação**, que são configurados nas propriedades do objeto main (possivelmente um menu, como no caso de nossa aplicação) como vemos aqui.

No caso do Android, minimamente, precisamos do Android Application Icon, do Android Portrait Launch Image e do Android Landscape Launch Image, no caso da Apple, basta inserir o Apple Application Icon e a Apple Launch Image.

Bem, estas foram algumas das características mais importantes sobre o Objeto Theme, o tratamento das imagens e a configuração das plataformas, durante o restante do curso apresentaremos algumas classes e propriedades específicas do desenvolvimento para Smart Devices, também podem aprofundar sobre estes temas no wiki do GeneXus.

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications