



Controls: Tables, Labels, Images Scaling & Live Editing

User Experience & User Interface

GeneXus 16

- 1 Labels
- 2 Live Editing
- 3 Tables
- 4 Images Scaling

Neste vídeo veremos as características das etiquetas de atributos e variáveis, então apresentaremos a ferramenta Live Editing, continuaremos com o uso das tabelas e para finalizar falaremos sobre o dimensionamento das imagens e suas variantes.

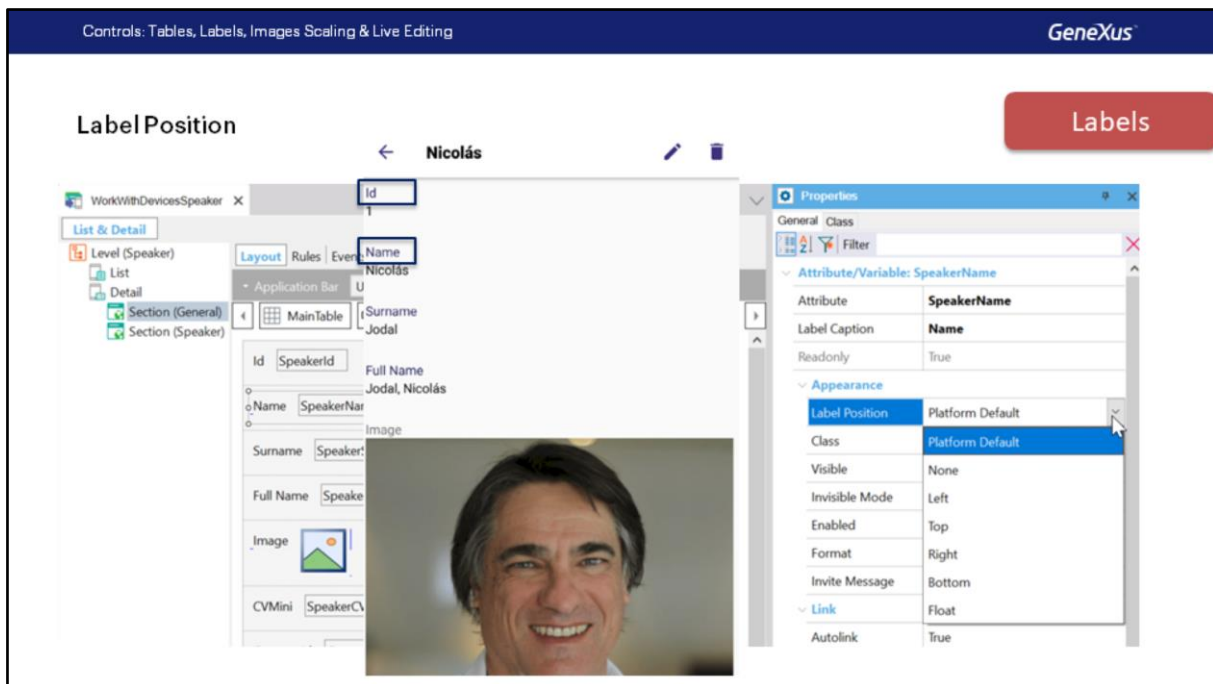
SpeakerDetail



Continuaremos com o desenvolvimento de nossa app.

Vamos lembrar como era vista a tela de Detalhe de um Palestrante na app que apresentamos no início do curso, corresponde à imagem que vemos à esquerda da tela, em nossa app esta mesma tela é vista como mostrado à direita.

Observemos que a tela claramente não tem o mesmo design, embora sejam parecidas. Então, vamos modificá-la para nos aproximarmos do design final, em particular nesta primeira parte vamos nos concentrar nas etiquetas.



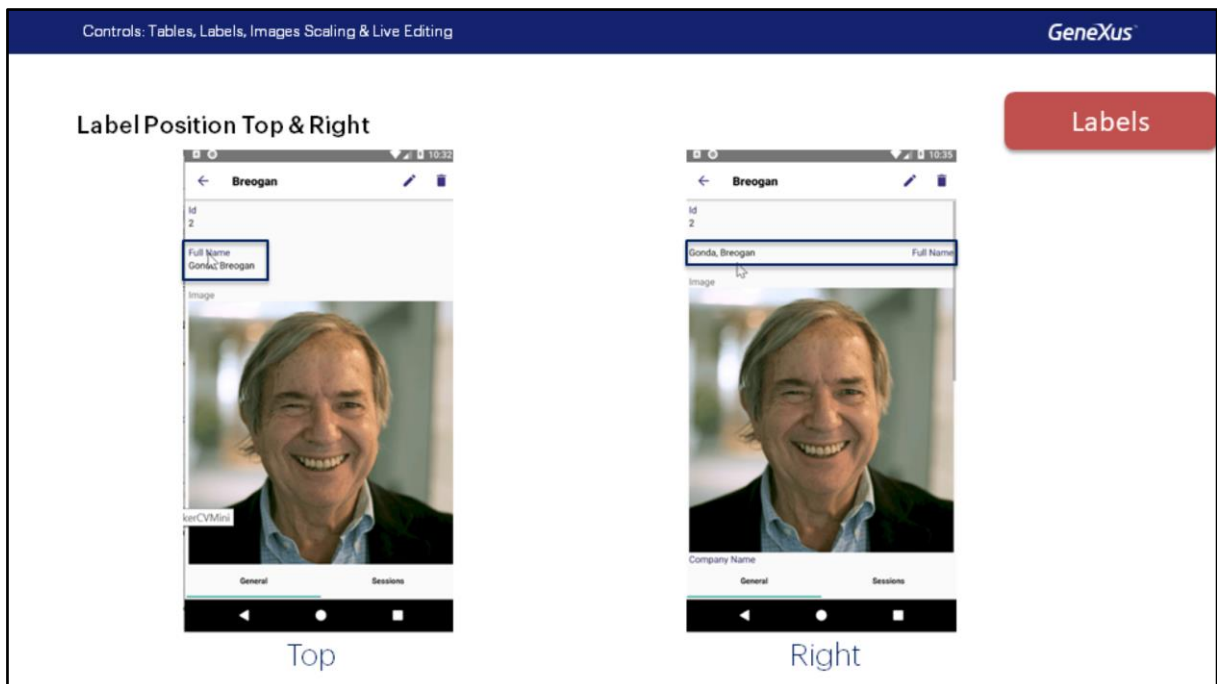
Em nossa App, as etiquetas dos atributos estão aparecendo acima do conteúdo, por exemplo, Id, Name.

Para cada atributo podemos configurar como serão vistas as etiquetas, as opções são:

- Platform Default: para Android, é o valor Top, acima do conteúdo, e para iOS é à esquerda.
- None: não mostra a etiqueta.
- Left: à esquerda do conteúdo
- Top: acima do conteúdo
- Right: à direita do conteúdo
- Bottom: abaixo do conteúdo
- Float: a etiqueta é mostrada na posição do conteúdo do atributo ou variável, quando o usuário começa a escrever, a etiqueta se move para cima do mesmo, como se estivesse flutuando. Isto só tem efeito em campos editáveis e em campos numéricos ou de tipo caractere, no restante dos tipos o valor adotado é Top. No caso do Android, o valor da etiqueta e o valor de Invite Message devem ser os mesmos, caso contrário, será exibido o valor de Invite Message.

Demo: Changing Labels Positions

Vamos ver tudo isso no GeneXus, assim testamos as diferentes maneiras de mostrar as etiquetas.



Abriremos o objeto que estávamos vendo, WorkWithDevicesSpeaker e nos posicionamos na seção General.

Revisamos o valor de Label Position, por exemplo, em SpeakerId, seu valor é Platform Default, o mesmo em SpeakerFullName.

Vejamos como a tela está sendo vista agora no emulador, a etiqueta está sendo mostrada acima do conteúdo, que é o valor Default do Android.

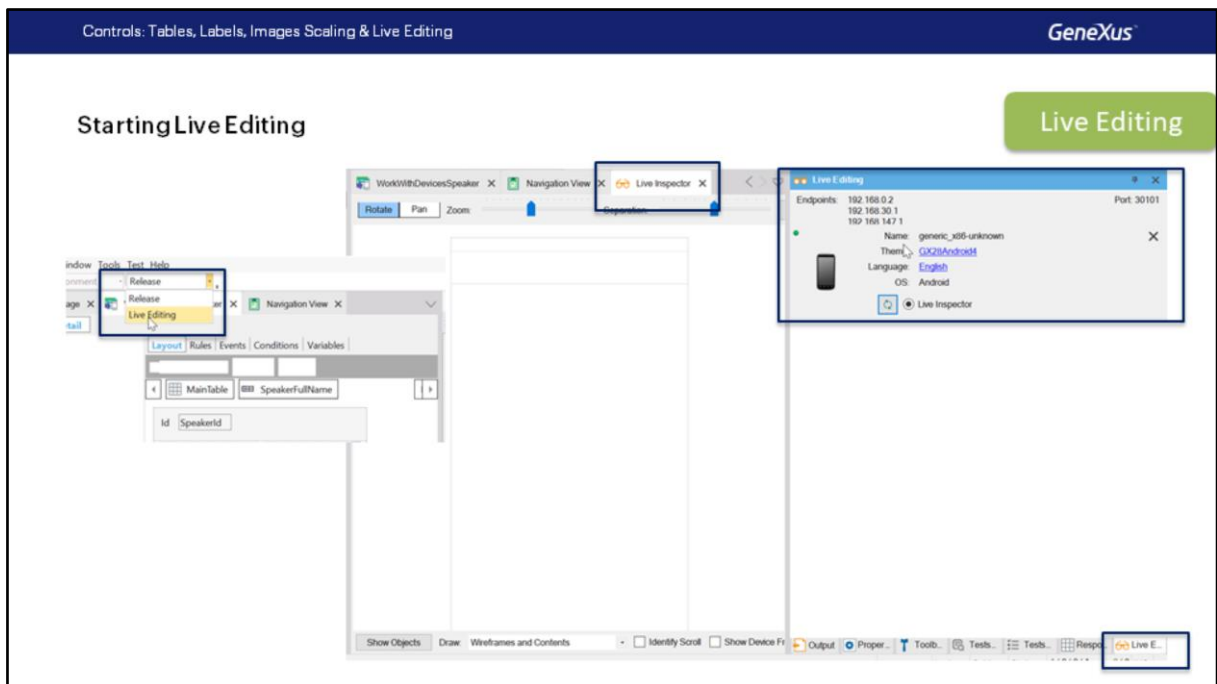
Alteraremos o valor para SpeakerFullName, utilizaremos por exemplo Right.

Bem, vamos executar a aplicação. pressionamos RUN.

Algo que vocês já notaram é todo o tempo necessário para especificar e compilar tudo para, em seguida, ver as alterações no emulador.

Já temos a aplicação instalada, vamos rever como ficou.

Entramos em Speakers, selecionamos um Palestrante e vemos que a etiqueta agora é exibida à direita do conteúdo, conforme havíamos especificado.



Voltemos ao GeneXus. Aproveitaremos para utilizar um recurso muito interessante, trata-se de Live Editing.

Esta ferramenta nos permitirá fazer alterações em Temas, Idiomas, Layouts e Eventos do Usuário de nossos painéis para Smart Devices e ver instantaneamente essas alterações na app no dispositivo, sem precisar compilar nada, e sem sequer que essas alterações devam estar gravadas.

Vamos prototipar em Modo Live Editing, apenas mudamos de Release para Live Editing e executamos a aplicação novamente.

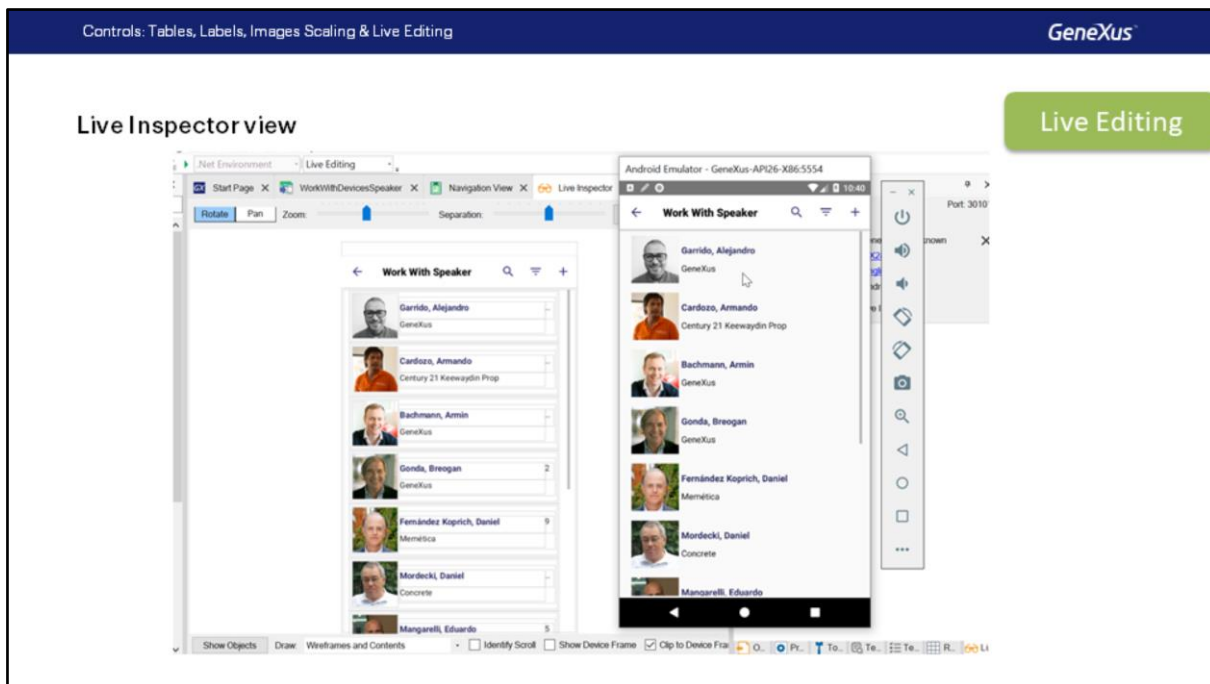
Neste caso, como é a primeira vez será necessário compilar a aplicação novamente, mas nas próximas alterações que fizermos, isto já não será mais necessário, simplesmente serão refletidas as alterações automaticamente no emulador e no IDE do GeneXus..

Já temos a aplicação pronta novamente no Emulador, só que agora habilitamos Live Editing.

No IDE podemos ver que foi habilitada uma aba Live Editing, lá temos as informações dos dispositivos que estão conectados ao Live Editing, para cada um podemos ver o nome do dispositivo, o tema que está sendo usado na aplicação, o idioma e o Sistema Operacional.

Também podemos ver que uma nova janela foi aberta, trata-se do Live Inspector, esta é

uma das principais vantagens do Live Editing, embora haja uma restrição e não possamos ver quando se está executando um objeto Menu For Smart Devices, o resto dos painéis, sim poderemos vê-los, tenham sido eles desenvolvidos com WorkWith For Smart Devices ou Panel For Smart Devices.



Aqui podemos ver a Lista de Palestrantes, e vemos que tudo o que vemos no emulador ou no dispositivo será refletido no IDE dentro da tela do Live Inspector.

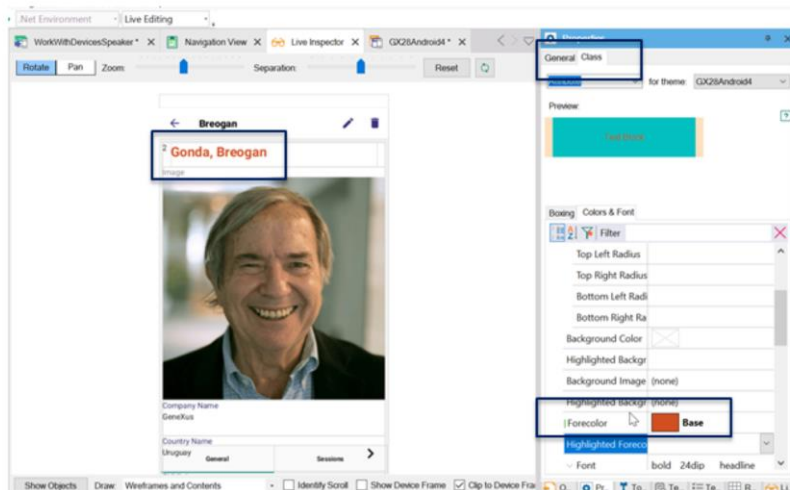
Vamos acessar um Palestrante, se selecionarmos um controle no Live Inspector, podemos ver suas propriedades, por exemplo, vamos para `SpeakerFullName` e agora alteramos o valor de `Label Position` para `Left`. e já vemos como fica, não foi necessário gravar nem compilar, nem executar a aplicação novamente. Se agora colocamos o valor `Bottom` novamente, o vemos imediatamente. Vamos deixar o valor `None`.

Vamos selecionar agora o atributo `SpeakerId`, ocultaremos também sua etiqueta, na verdade vamos deixá-lo oculto, o atributo diretamente. colocamos a propriedade `visible` em `False` e colocamos a propriedade `Invisible Mode` em `Collapse Space` para que, quando estiver oculto, o controle não ocupe espaço na tela.

Agora vamos mudar a classe para o atributo `SpeakerFullName`, vamos usar a classe `AttributeSessionTitle`, vamos ver que já temos a mudança no IDE.

Editing Object & Theme with Live Inspector

Live Editing



Além da possibilidade de alterar as propriedades dos atributos, na aba Class podemos editar diretamente o Tema, vamos para Colors & Fonts, e usamos uma cor mais marcante, escolhemos a cor Base na paleta, pronto, já podemos ver a mudança.

Vamos desfazer essa mudança, pois foi apenas para mostrar como usá-la.

Agora vamos atribuir a classe AttributeSpeakerNameT, que é a classe que o designer nos indicou.

Vamos agora diretamente para o objeto, aqui também podemos fazer alterações e vê-las refletidas instantaneamente, de fato, veja que já tem os valores que acabamos de colocar no Live Inspector.

Outra coisa que podemos fazer é, por exemplo, mover um controle, levamos o atributo SpeakerImage acima de SpeakerFullName e o deixamos sem Etiqueta.

Vamos remover a etiqueta de CompanyName e atribuir-lhe a classe AttributeSpeakerBioT. Agora para CountryName, fazemos o mesmo, removemos a etiqueta e atribuímos a mesma classe.

Levamos o atributo SpeakerCVMini acima, abaixo de SpeakerFullName. Também removemos a etiqueta e atribuímos a mesma classe.

Marquemos agora os atributos SpeakerAddress, SpeakerEmail e SpeakerPhone, mantendo pressionado Control podemos selecionar vários controles.

Atribuímos Label Position TOP, vejamos como fica no emulador.

OK, faltou atribuir a classe, atribuímos a classe, a mesma classe que usamos antes.

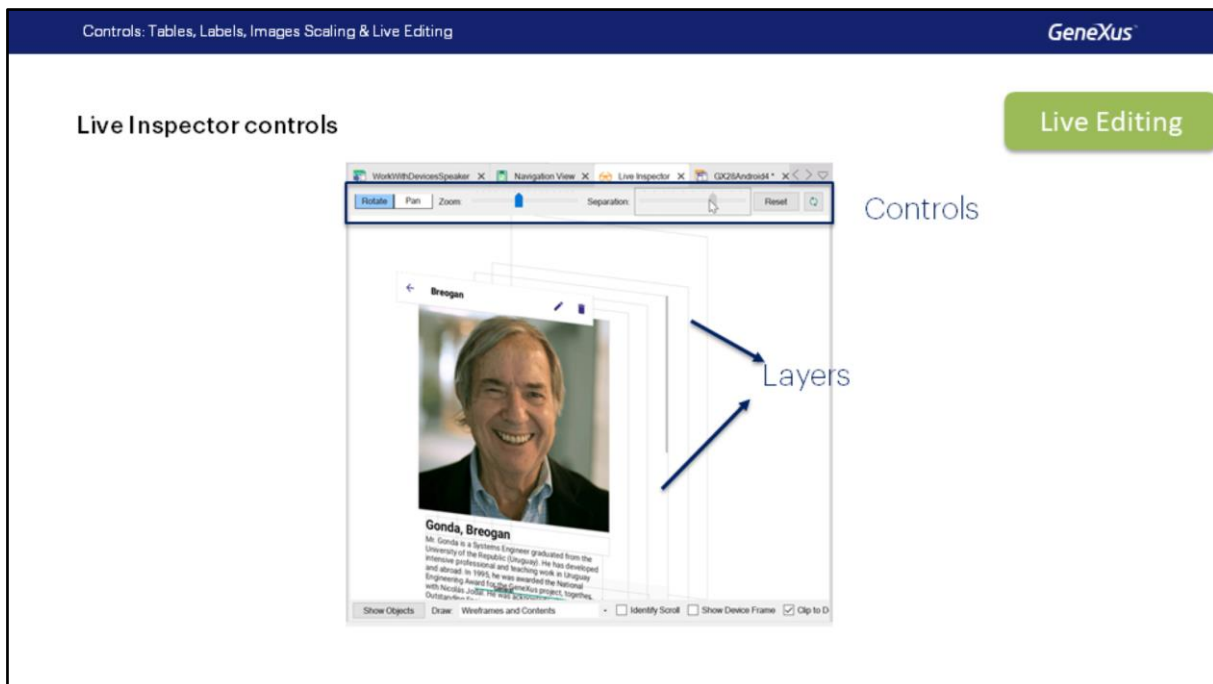
Vejamos novamente como ficou, OK. agora sim.

Vamos adicionar um título, vamos para a Tool Box e adicionamos um controle do tipo

TextBlock acima de SpeakerAddress, vamos colocar Caption Contact e a classe

TextBlockTitleWidgetM, vamos revisar, está perfeito.

Agora sim, vamos apenas gravar as mudanças porque estamos satisfeitos com o design.



Agora vamos ver alguns outros recursos da tela do Live Inspector.

Por exemplo, se clicarmos e movermos o mouse, poderemos girar a imagem em qualquer direção e percebemos que algumas camadas serão vistas quando fizermos isto.

Estas camadas nos permitem selecioná-las de uma maneira muito mais simples, uma vez que elas estão geralmente sobrepostas, quando são seções ou tabelas.

com este controle podemos aumentar ou diminuir o espaço entre as camadas, com Reset voltamos à visualização padrão, com este controle podemos aumentar ou diminuir o Zoom no que estamos vendo e se selecionarmos Pan, em vez de Rotate, podemos mover o conteúdo com o mouse, sem Girá-lo.

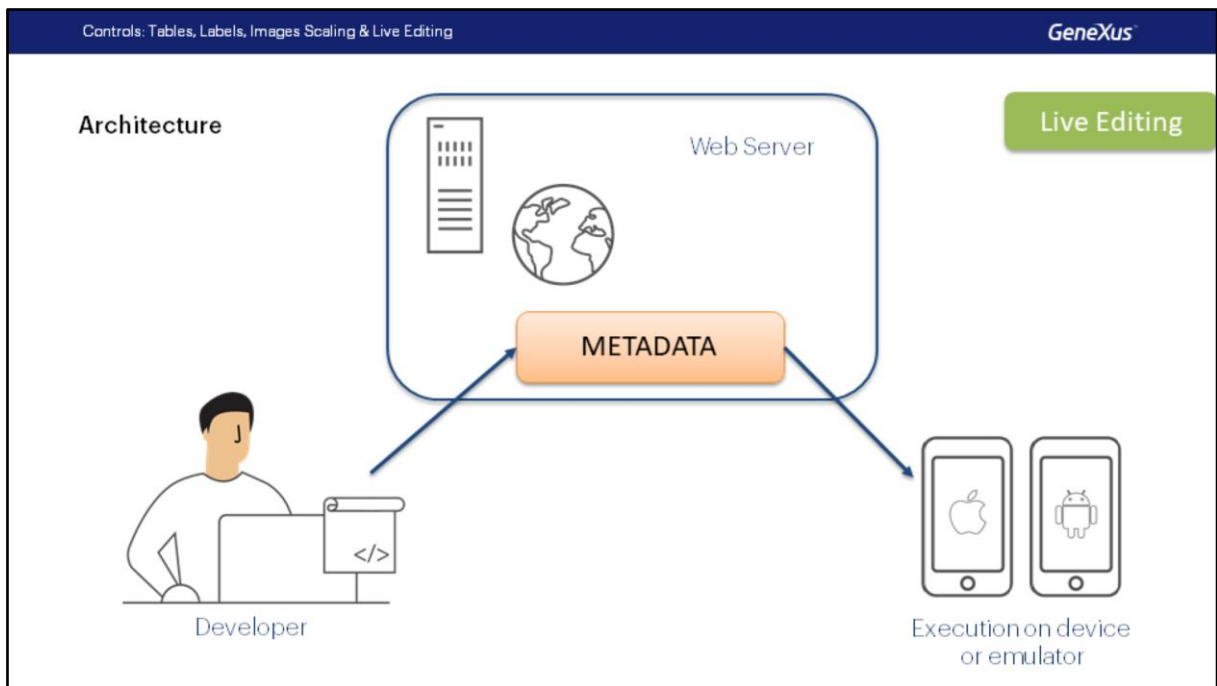
Bem, isto foi tudo, Vamos voltar agora para a apresentação, então revisamos outros assuntos relacionados ao Live Editing.

Live Editing Restrictions**Live Editing**

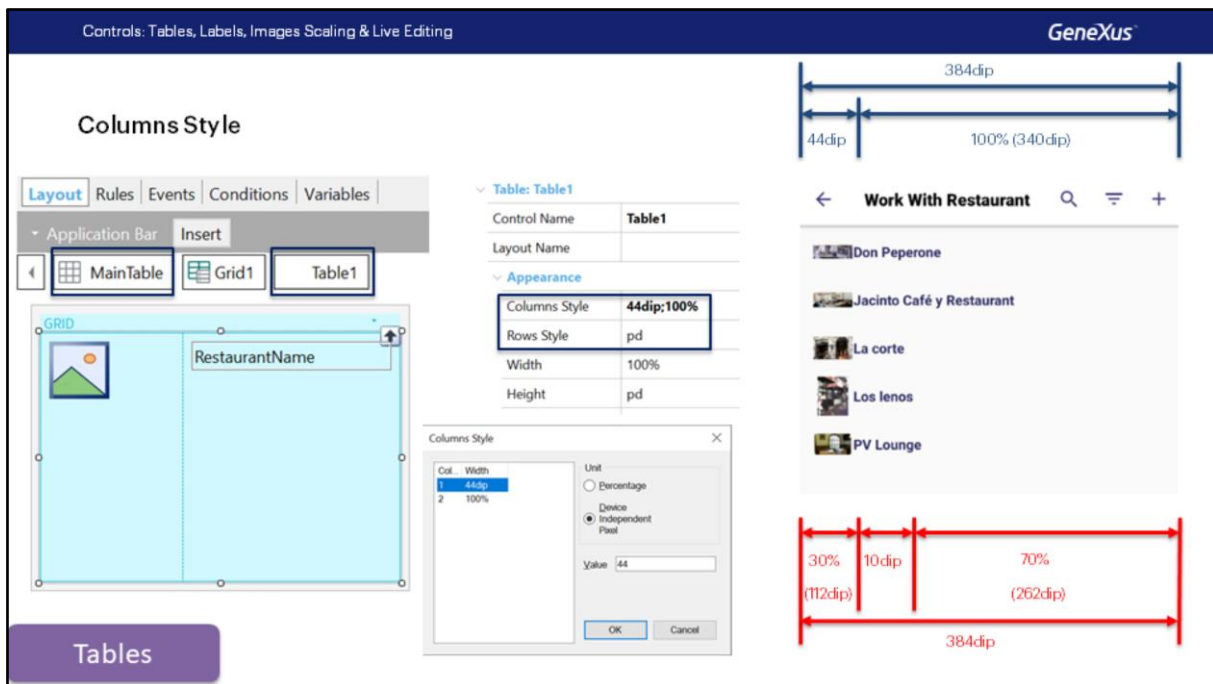
- Object Menu for Smart Devices is not supported.
- Server-side events (such as Start, Refresh and Load) cannot be tested with Live Editing because they are executed in the server-side (not in the device).
- Live Editing only works with the compiled application. Knowledge Base Navigator (KBN) is not supported.
- Live Inspector doesn't work in Windows 7.

As restrições para o uso do Live Editing são:

- Não funciona em objetos do tipo Menu for Smart Devices.
- As alterações nos eventos Server Side (do lado do servidor) como Start, Refresh e Load não têm efeito, portanto, é necessário recompilar a aplicação.
- Só funciona quando temos uma aplicação compilada, não podemos usá-lo quando prototipamos com o KBN.
- Não funciona no Windows 7.



Como este recurso funciona? Quando usamos o Live Editing, o Servidor ficará “escutando” as mudanças que fizemos e replicará estas mudanças nos Metadados acessados pelo dispositivo móvel (no nosso caso, o emulador) e que é usado para desenhar a tela no dispositivo e definir seu comportamento.



Vamos continuar com o design, agora focando nossa atenção nas Tabelas.

Nesta seção, usaremos o WorkWithDevicesRestaurant, se ainda não aplicou o padrão a esta transação pode fazê-lo, também adicionamos uma opção ao menu para acessá-lo, como fizemos com Palestrantes e Países.

O trabalho com as tabelas será central nas aplicações para Smart Devices. Todo layout, mesmo se vazio, tem uma tabela raiz, chamada MainTable. Por sua vez, todo grid irá definir uma tabela para conter os controles de cada linha a ser carregada. Neste caso, a tabela tem duas colunas e uma linha. Vejamos as duas propriedades: **Columns Style** e **Rows Style**, que permitem definir o tamanho que ocupará cada coluna e cada linha da tabela.

Observemos as colunas. Seus tamanhos podem ser especificados em duas unidades: porcentagem ou dips (**Device Independent Pixel**).

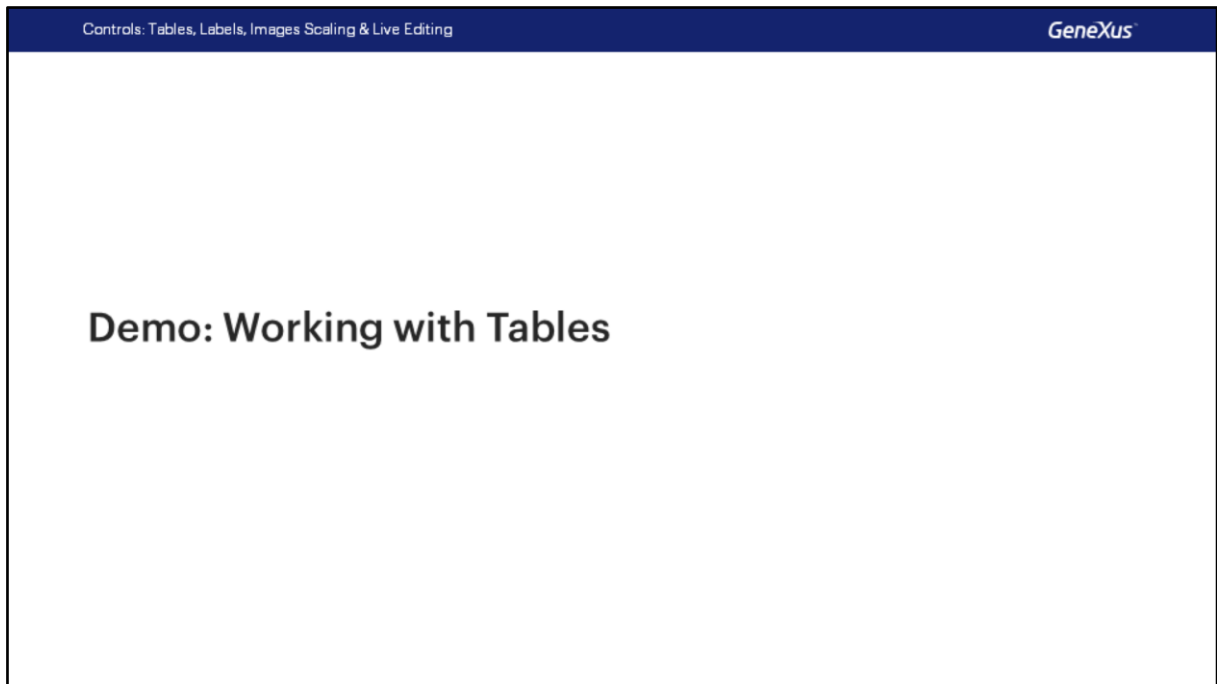
O Device Independent Pixel corresponde a uma abstração de um pixel que em seguida uma aplicação converte em pixels físicos, o que permite dimensionar para diferentes tamanhos de tela. O dip para cada plataforma tem um número diferente de pixels.

As porcentagens são relativas ao valor resultante da subtração da largura total, os valores fixos (em dips). Assim, se a largura total for 384 dips, e houver uma coluna de 44 dips, o

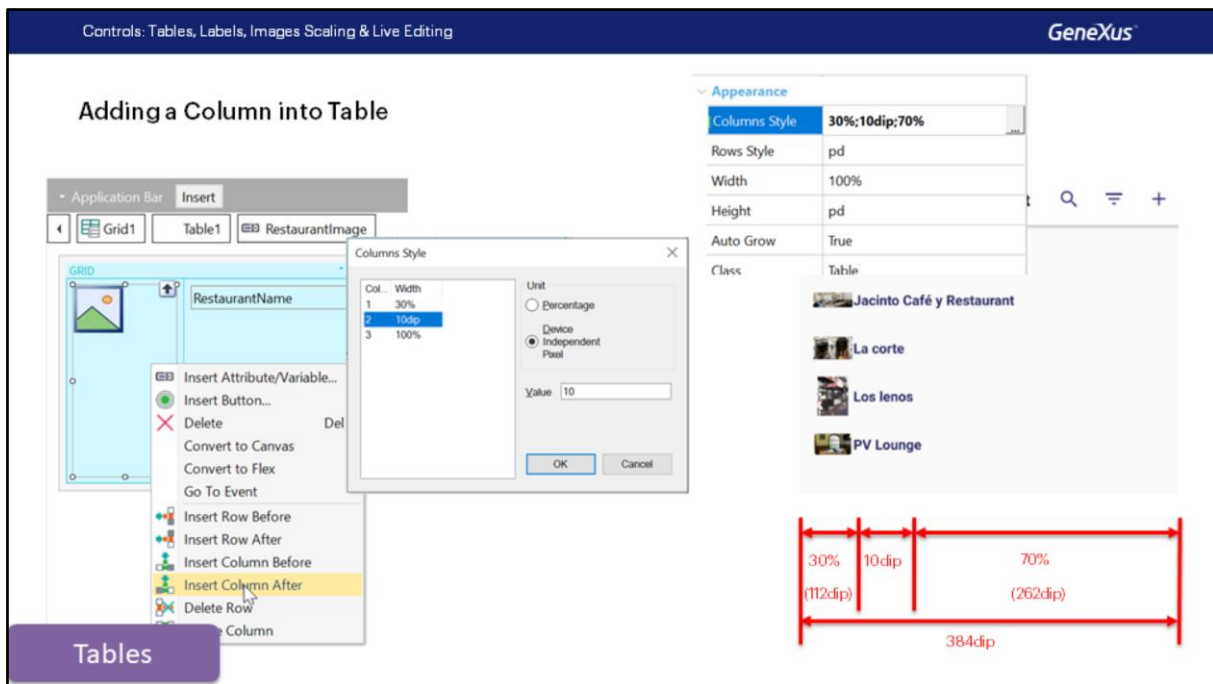
valor de 100% para a próxima significa que essa segunda coluna ocupará os 100% restantes, isto é, $384 - 44 = 340$ dips.

Se tivéssemos três colunas, a primeira com 30%, a segunda com um valor fixo de 10 dips e a terceira com 70%, os valores que assumirão a primeira e a terceira são obtidos aplicando-se essas porcentagens ao valor resultante da subtração da soma dos valores fixos (aqui apenas um, 10 dips) da largura da tabela. A segunda coluna com 10 dips pode ser usada como um espaço em branco entre a primeira e a terceira coluna.

Devido a isto, a quantidade de espaço disponível para a primeira e terceira colunas é: $384 - 10 = 374$ dips, portanto a primeira coluna terá 30% de 374 o que equivale a 112 dips e a terceira os 70% restantes, que são 262 dips.



Vejamos tudo isto no GeneXus.

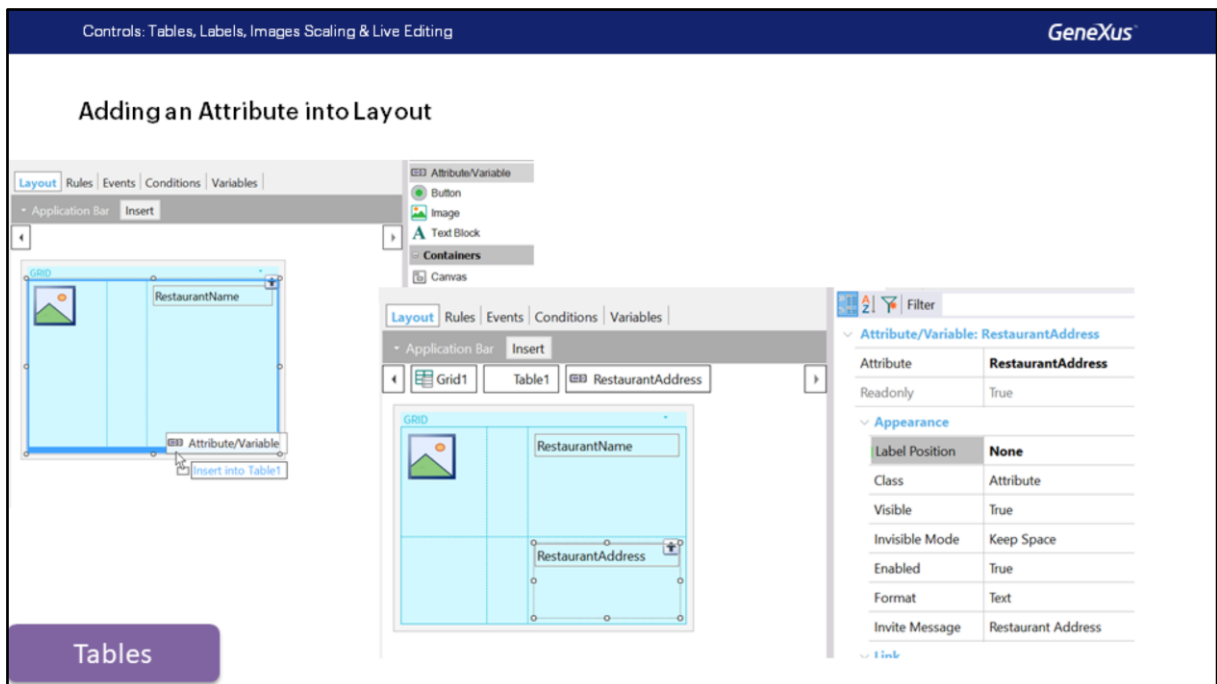


Vamos acessar o objeto WorkWithdevicesRestaurant e vamos para o List.

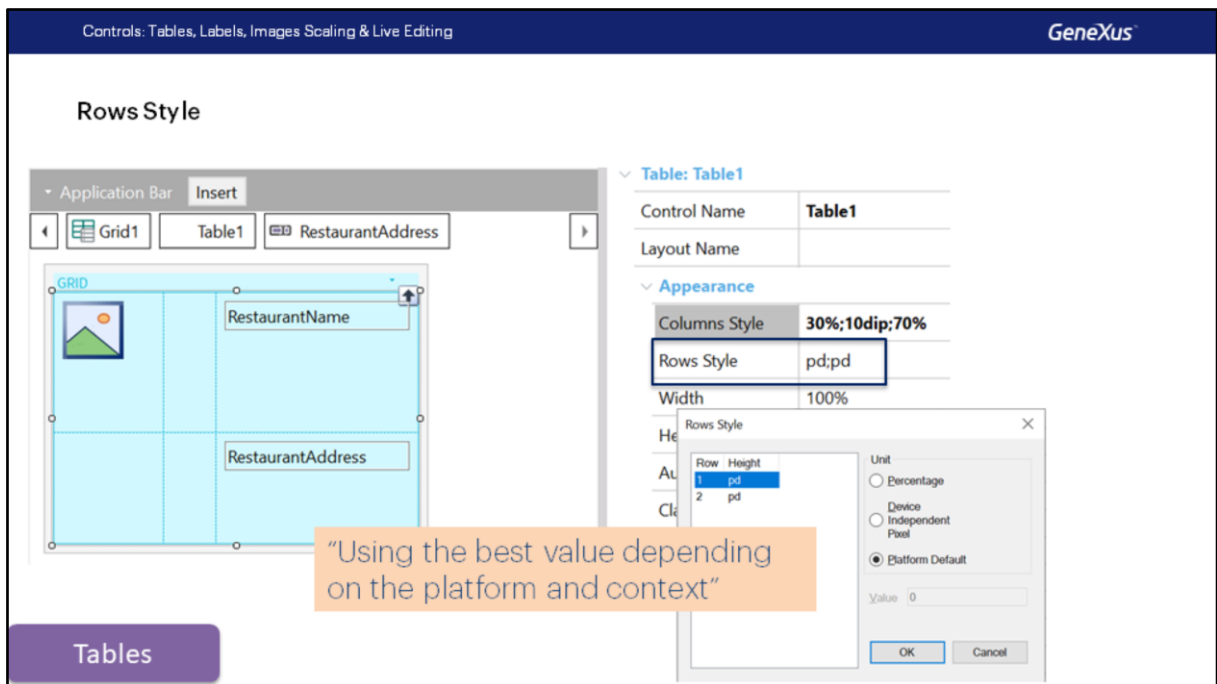
Aqui podemos ver a Main Table da qual falamos recentemente e dentro do grid podemos ver a tabela com o nome Table1 que atualmente possui 2 colunas, uma de 44 dips e outra de 100%, como vimos na apresentação.

Então, vamos inserir uma coluna adicional usando o botão direito na tabela do grid e a adicionamos depois onde estamos, RestaurantImage.

Vamos especificar o valor do Columns Style como vimos, vamos colocar 30% para a primeira coluna, 10 dip para a segunda e 70% para a terceira, vamos ver a aplicação em execução assim vemos o resultado, estamos usando o Live Editing para que possamos ver diretamente, e aqui vemos que mudou, agora temos um espaço entre o nome e a imagem.



Vamos adicionar o atributo RestaurantAddress ao grid, vamos remover sua etiqueta (colocamos a Label Position em None), se vemos agora no Live Inspector, não vemos a mudança, isto é porque estamos fazendo uma alteração, uma modificação, em um evento Server Side (o load) então é necessário recompilar a aplicação, portanto pressionamos Run novamente.



Enquanto a aplicação é gerada e compilada vamos revisar as linhas da tabela, vemos que a unidade pd, Platform Default é adicionada. Esse valor difere de plataforma para plataforma e, para uma mesma plataforma, também depende do conteúdo da célula, por exemplo, se o campo tem label ou não, e se tiver, se a etiqueta será exibida na parte superior ou à esquerda, ou se estamos em um layout no modo Edit ou no modo View.

O valor pd Corresponde a: “Usar o Melhor Valor Dependendo da Plataforma e do Contexto”.

Por exemplo, para Android, com Label Position = Top, corresponde a 64 dips, enquanto no iOS a 53 dips.

Vejamos como ficou, bem, já vemos o Address, vamos mudar a classe, vamos criar uma classe AttributeRestaurantAddress e vamos configurar a classe, removemos Autolink (False), não queremos que tenha o ícone para ir para o endereço, e vamos colocar em Font Weight Bold, pronto.

Já temos o design, gravamos e continuamos com a apresentação.

Different values for Image Scale Type

No Scale	Respects the original size of the image, independently of the area control size.
Tile	The image is not scaled. It is repeated horizontally and vertically to fill the control size.
Fill	The image is scaled in width and height to fill the whole size of the control area.
Fill Keeping Aspect Ratio	The image is made bigger or smaller in width and height to fill the entire size of the control area, but keeping the aspect of the image. For example, if the image size is 100x200, and the control size is 50 x 50, then the image size is converted to 50 x 100.
Fit	The image scales in width and height to see it at all, and keeping the aspect of the image. For example, if the image is 100x200, and the control is 50 x 50, then the image is converted to 25 x 50. This is the default value.

Vejamos algumas propriedades das imagens, especificamente os diferentes valores que temos para o dimensionamento das imagens, estas são:

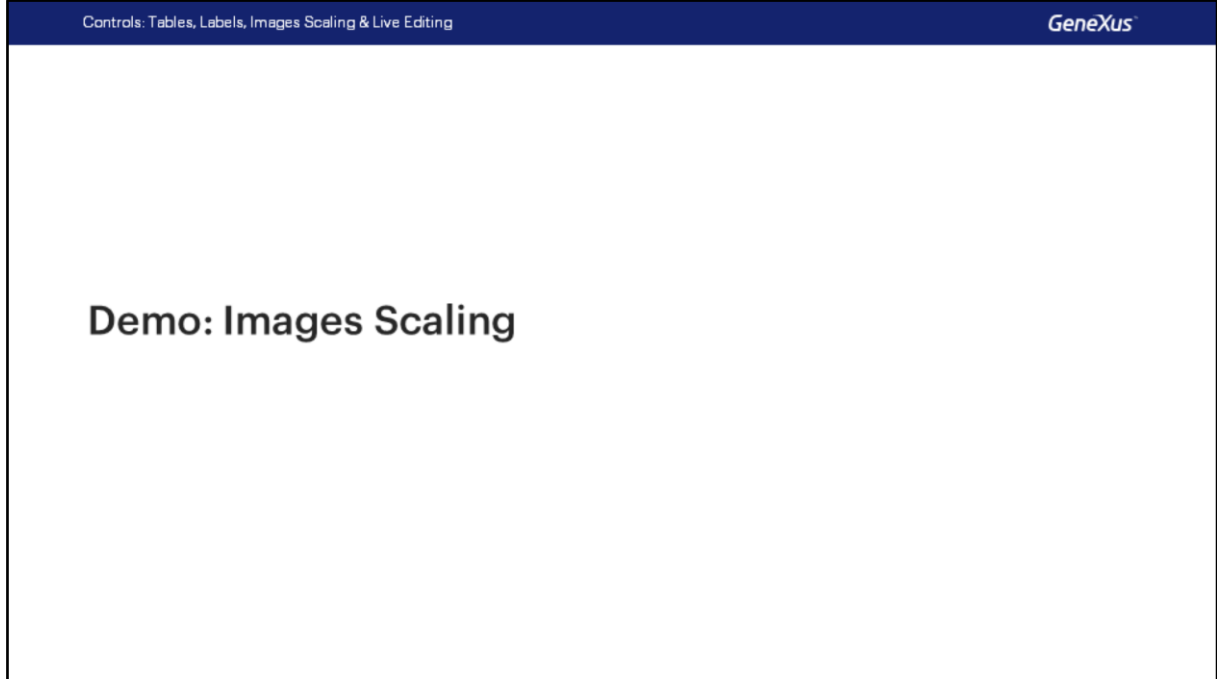
No Scale: o tamanho original da imagem será respeitado, independentemente do tamanho da área onde o controle está.

Tile: neste caso, as imagens não são dimensionadas e a imagem será repetida até que o tamanho do controle seja preenchido.

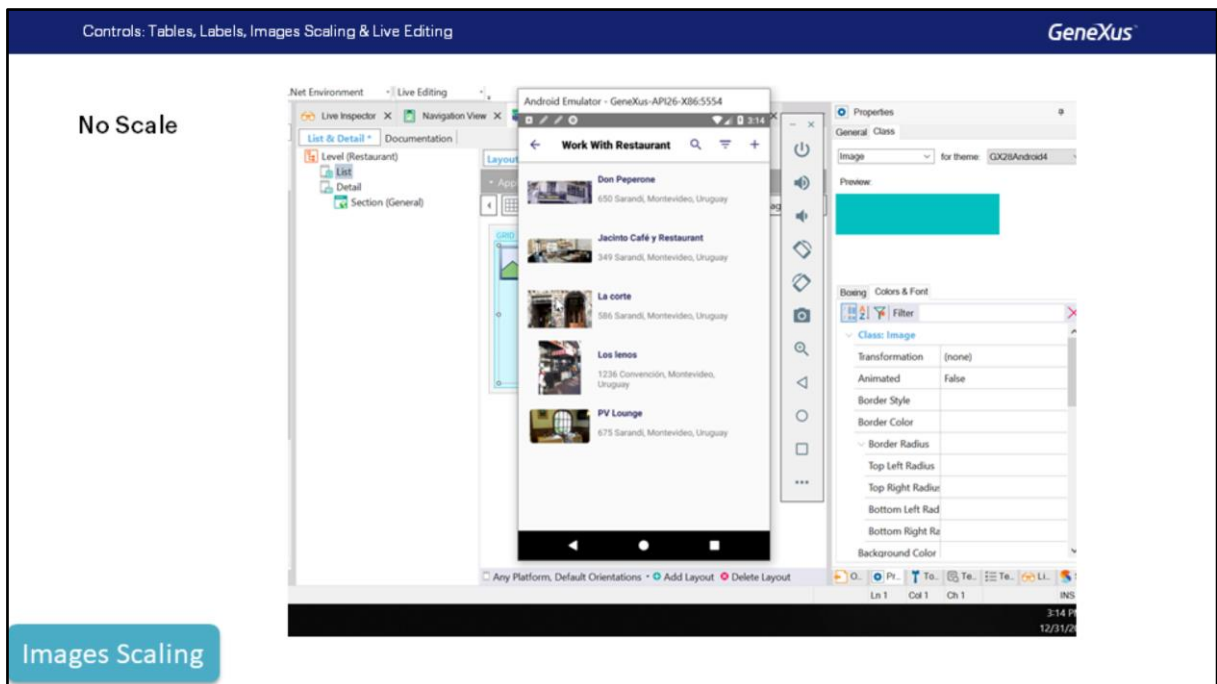
Fill: a imagem irá redimensionar em altura e largura até ocupar o tamanho da área onde está o controle.

Fill Keeping Aspecto Ratio: é semelhante ao Fill, mas a imagem sempre manterá as mesmas proporções de largura e altura, desta forma a imagem não se deforma. Por exemplo, se o tamanho da imagem for 100x200 px e o tamanho da área de controle for 50x50, a imagem será convertida em 50x100 px.

Fit: a imagem será dimensionada em altura e largura para que possa ser vista integralmente e mantendo as proporções e a aparência. Por exemplo, se a imagem for 100x200 px e o tamanho da área de controle for 50x50, a imagem será convertida para 25x50. Este é o valor padrão que é assumido.



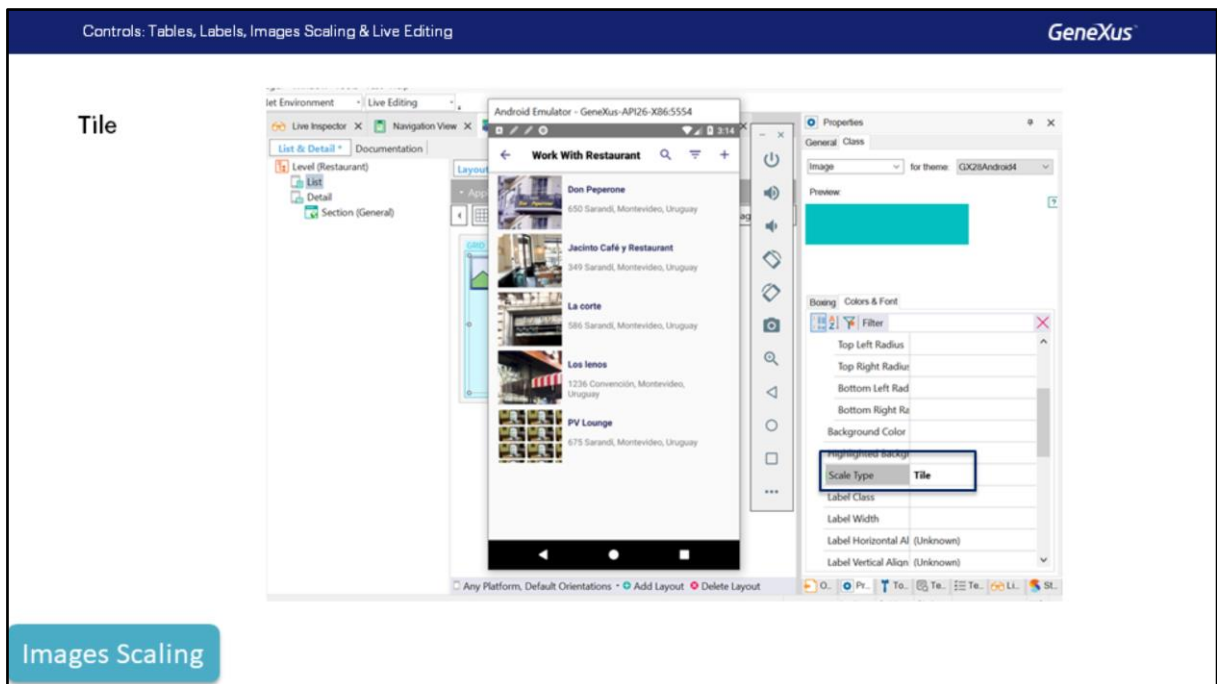
Voltemos ao GeneXus e veremos isto aproveitando Live Editing.



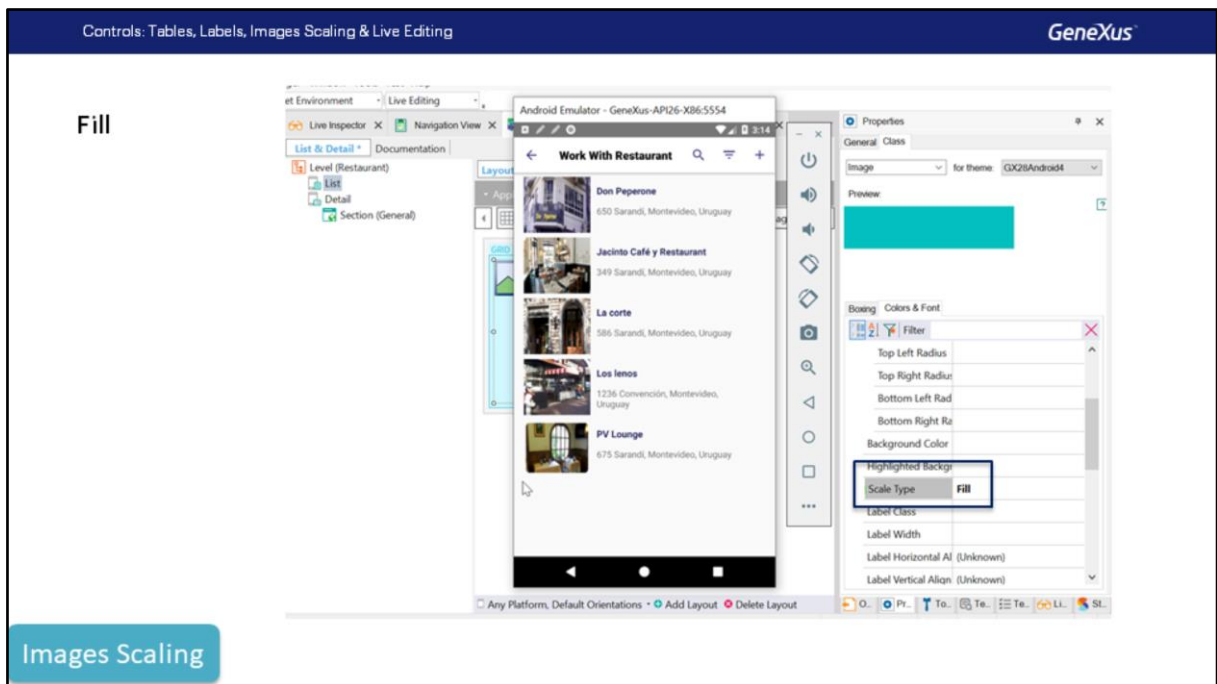
Assim é como se está vendo o List de Restaurants, observe que as imagens não estão ficando muito bonitas e isto é devido ao dimensionamento das imagens, vamos tentar consertar este painel.

A primeira coisa que faremos é fazer com que esta imagem ocupe 2 linhas, esta é uma propriedade Row Span do controle, e indicaremos que ocupe duas linhas, vamos excluir a coluna que foi adicionada automaticamente.

Agora note que tínhamos uma coluna de 30%, outra de 10 dips e 70% a restante. Vejamos como está ficando, note que as imagens estão um pouco distorcidas, isto é justamente por causa da propriedade Scale Type das imagens, a classe de RestaurantImage é a classe Image, e vamos ver que em Scale Type não temos nenhum Valor. Vamos alterando estes valores para ver como vão se modificando em execução. NO Scale, vemos que continua igual.

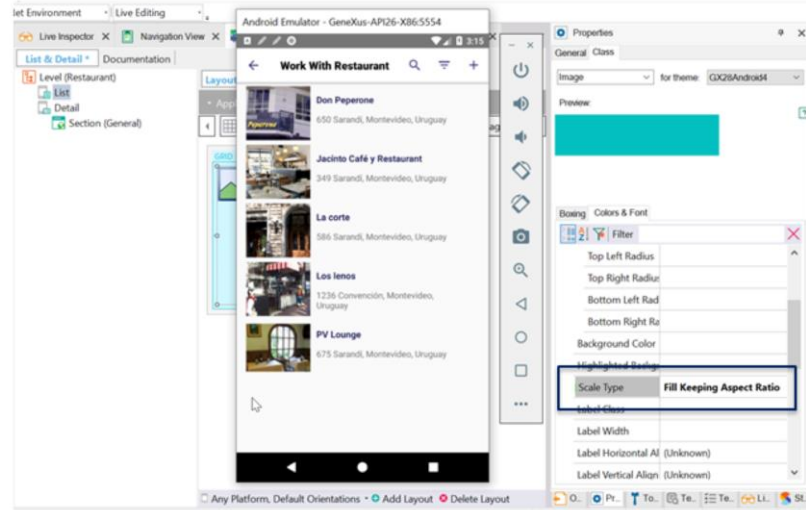


Em Tile, não apreciaremos esse efeito. Bem, neste caso se (a última imagem) esta imagem é menor que o tamanho (da área de controle), portanto a imagem é repetida para preencher o espaço disponível.



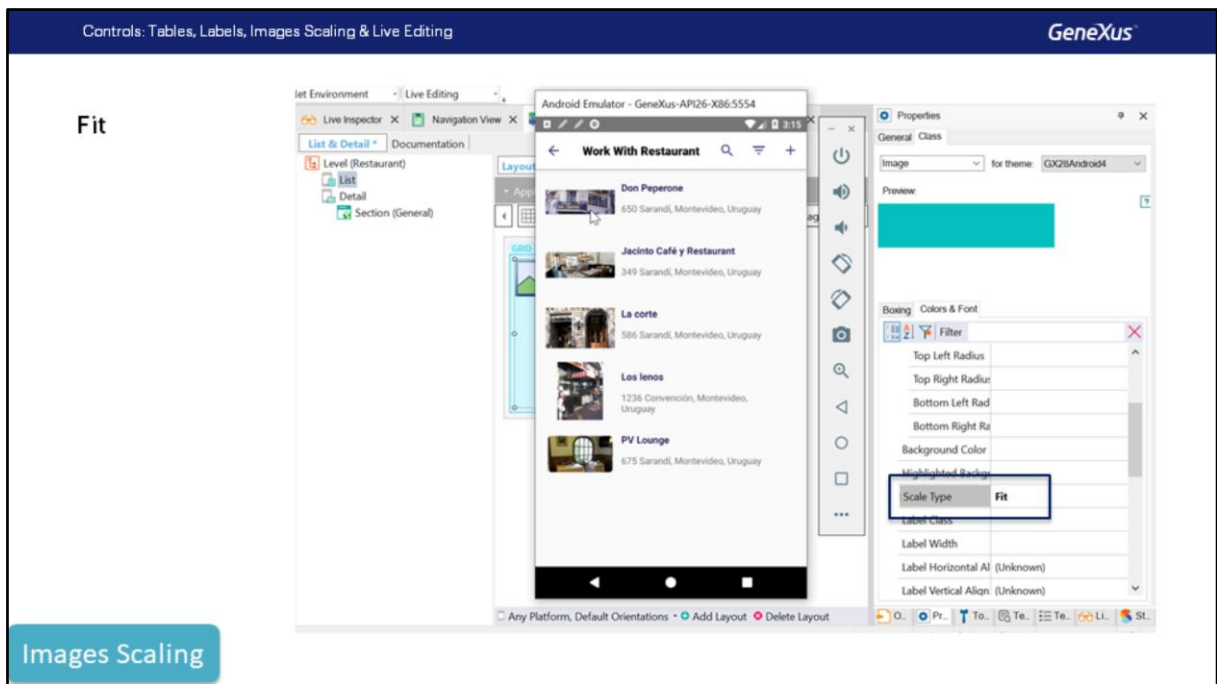
Vejamos com Fill, observe que, neste caso, a imagem preenche todo o espaço disponível, mas se deforma.

Fill Keeping Aspect Ratio



Images Scaling

Vejamos Fill Keeping Aspect Ratio, observe que, agora sim, a imagem preenche todo o espaço disponível, mas a aparência da imagem é mantida.



E Fit, que é o último valor que nos resta, veja que preenche, mas não ocupará todo o espaço disponível porque mantém a proporção quando preenche, mas não o faz completando todo o espaço disponível.

Vamos deixar então o melhor que foi visto, que foi Fill Keeping Aspect Ratio. Perfeito.

Label Position	Live Editing	Tables	Images Scaling
• Platform Default • None • Right • Left • Top • Bottom • Float	• Live Inspector • Using Restrictions • Architecture	• Columns Styles • Rows Styles • Measure Units	• No Scale • Tile • Fill • Fill Keeping Aspect Ratio • Fit

Vamos rever tudo o que vimos neste vídeo.

- Vimos as diferentes opções para Label Position: Platform Default, None, Right, Left, Top, Bottom e Float. Este último somente quando é em modo edição.
- Aprendemos a utilizar Live Editing, usamos a ferramenta Live Inspector e falamos sobre as restrições para seu uso e a arquitetura desta ferramenta.
- Sobre Tabelas, vimos como especificar os estilos das colunas e das linhas. Vimos as unidades de medida DIP, Percentagens e o valor pd, Platform Default, para Linhas.
- Por último, falamos sobre o dimensionamento de imagens e vimos os diferentes valores: No Scale, Tile, Fill, Fill Keeping Aspect Ratio e Fit.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications