

GeneXus™
The power of doing.

Controls: Tables, Labels, Images Scaling & Live Editing

User Experience & User Interface

GeneXus™ 16

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

En este video vamos a ver las particularidades las etiquetas de atributos y variables, luego presentaremos la herramienta Live Editing, continuaremos con el uso de las tablas y para finalizar hablaremos sobre el escalado de imágenes y sus variantes.

Speaker Detail

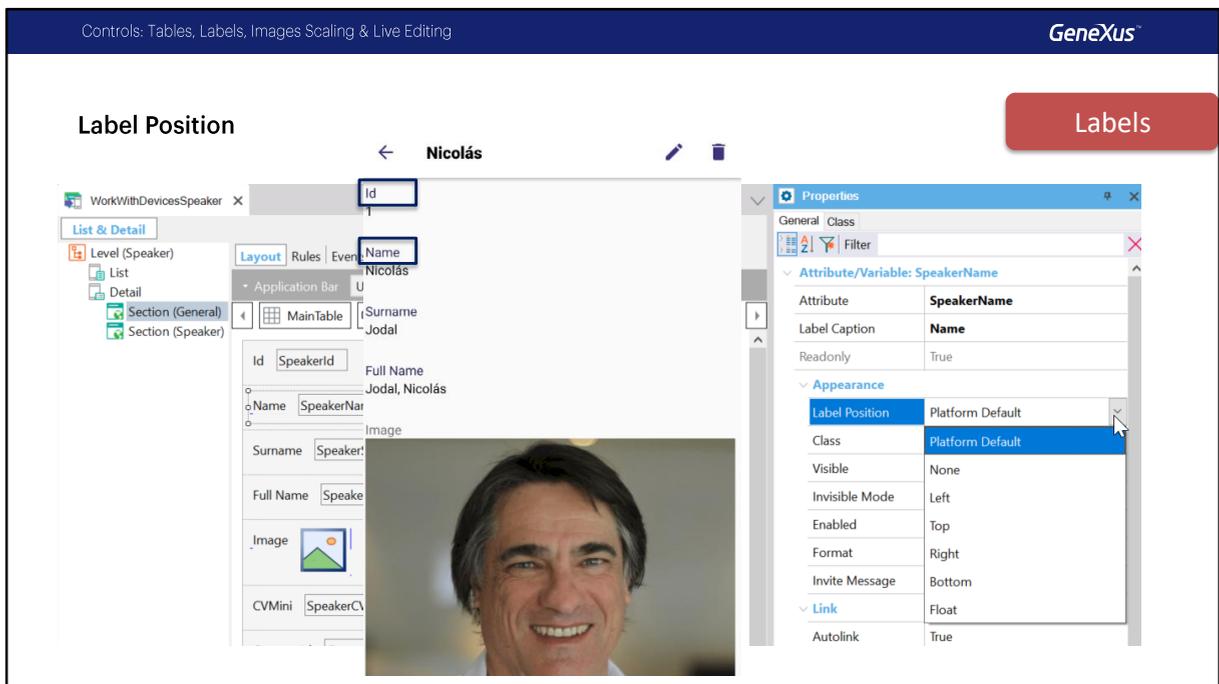


Vamos a continuar con el desarrollo de nuestra app.

Vamos a recordar como se veía la pantalla de Detalle de un Orador en la app que presentamos al inicio del curso, se corresponde a la imagen que vemos a la izquierda de la pantalla, en nuestra app esta misma pantalla se ve como se muestra a la derecha.

Notemos que la pantalla claramente no tiene el mismo diseño aunque son parecidas.

Luego la vamos a modificar para acercarnos mas al diseño final, en particular en esta primera parte nos vamos a concentrarnos en la las etiquetas.



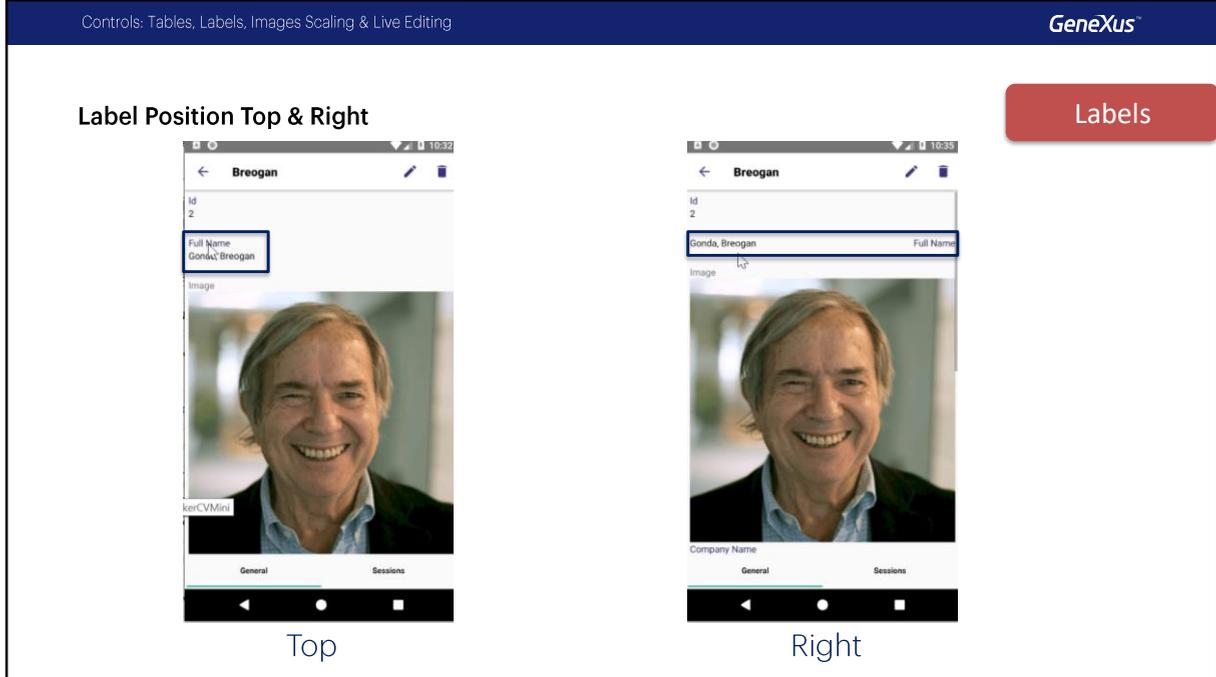
En nuestra App, las etiquetas de los atributos están apareciendo arriba del contenido, por ejemplo Id, Name.

Para cada atributo podemos configurar como se verán las etiquetas, las opciones son:

- **Platform Default:** para Android es el valor Top, arriba del contenido y para iOS es a la izquierda.
- **None:** no muestra la etiqueta.
- **Left:** a la izquierda del contenido
- **Top:** arriba del contenido
- **Right:** a la derecha del contenido
- **Bottom:** debajo del contenido
- **Float:** la etiqueta se muestra en la posición del contenido del atributo o variable, cuando el usuario comienza a escribir la etiqueta se mueve hacia arriba del mismo como si flotara. Esto solo tiene efecto en campos editables y en campos numéricos o de tipo carácter, en el resto de los tipos se adopta el valor Top. En el caso de Android, el valor de la etiqueta y el valor de Invite Message deben ser los mismos, de lo contrario se mostrara el valor de Invite Message.

Demo: Changing Labels Positions

Vamos a ver todo esto en GeneXus así probamos las distintas maneras de mostrar las etiquetas.



Abriremos el objeto que estábamos viendo, WorkWithDevicesSpeaker y nos posicionamos sobre la sección General.

Revisamos el valor de Label Position, por ejemplo en SpeakerId su valor es Platform Default, lo mismo en SpeakerFullName.

Veamos como se esta viendo la pantalla ahora en el emulador, la etiqueta se esta mostrando arriba del contenido, que es el valor Default de Android.

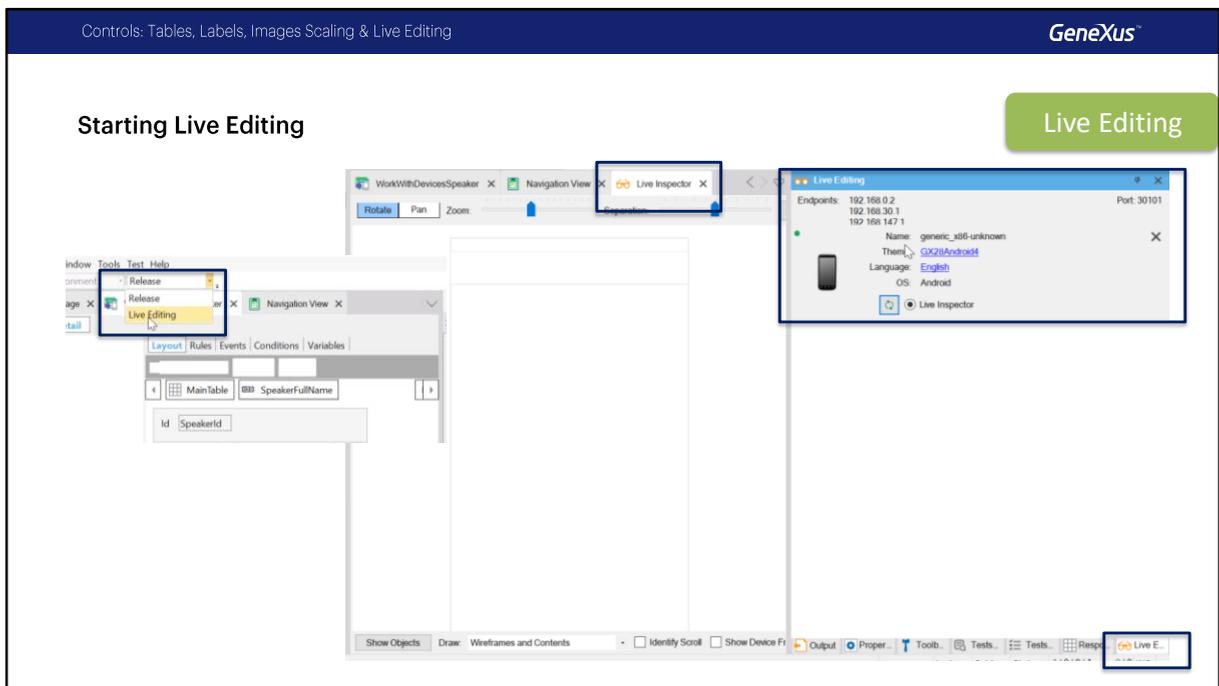
Cambiaremos el valor para SpeakerFullName, utilizaremos por ejemplo Right.

Bien, vamos a correr la aplicación. presionamos RUN.

Algo que ustedes ya habrán notado es todo el tiempo que lleva especificar y compilar todo para luego ver los cambios en el emulador.

Ya tenemos instalada la aplicación, vamos a revisar como quedó.

Ingresamos en Speakers, seleccionamos un Orador y ahí vemos que se muestra ahora la etiqueta a la derecha del contenido como habíamos especificado.



Vamos a volver a GeneXus. Aprovecharemos para utilizar una característica muy interesante, se trata de Live Editing.

Esta herramienta nos va a permitir realizar modificaciones en Temas, Lenguajes, Layouts y Eventos de Usuario de nuestros paneles para Smart Devices y ver instantáneamente esos cambios en la app en el dispositivo, sin tener que compilar nada, y sin que siquiera esos cambios deban estar grabados.

Vamos a prototipar en Modo Live Editing, simplemente cambiamos Release por Live Editing y corremos nuevamente la aplicación.

En este caso, como es la primera vez va a ser necesario compilar nuevamente la aplicación, pero los próximos cambios que efectuemos esto ya no será necesario, simplemente se verán reflejados los cambios automáticamente en el emulador y en el IDE de GeneXus.

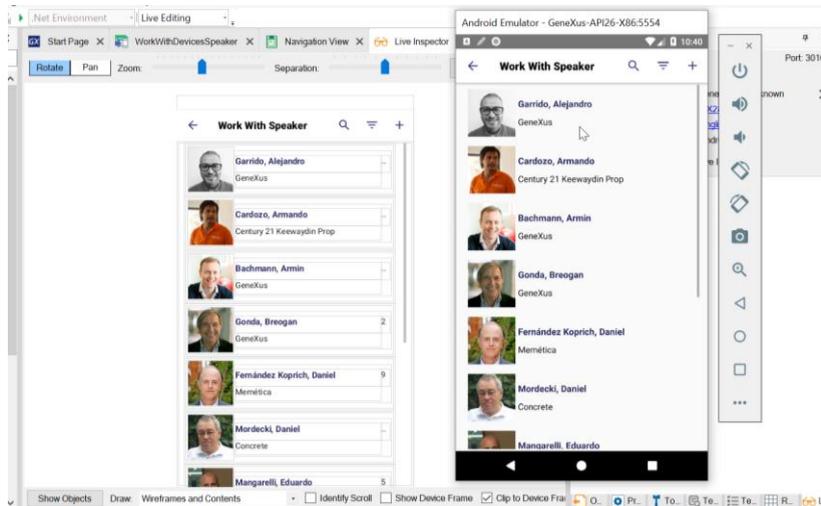
Ya tenemos lista la aplicación nuevamente en el Emulador, solo que ahora habilitamos Live Editing.

En el IDE podemos ver que se habilitó una solapa Live Editing, ahí tenemos la información de los dispositivos que se encuentran conectados a Live Editing, para cada uno podemos ver el nombre del dispositivo, el tema que se está utilizando en la aplicación, el lenguaje y el Sistema Operativo.

También podemos ver que se abrió una ventana nueva, se trata de Live Inspector, esta es una de las ventajas principales de Live Editing, si bien hay una restricción y no podemos ver cuando se está ejecutando un objeto Menu For Smart Devices, el resto de paneles si podremos verlos, ya sea que hayan sido desarrollados con WorkWith For Smart Devices o Panel For Smart Devices.

Live Inspector view

Live Editing

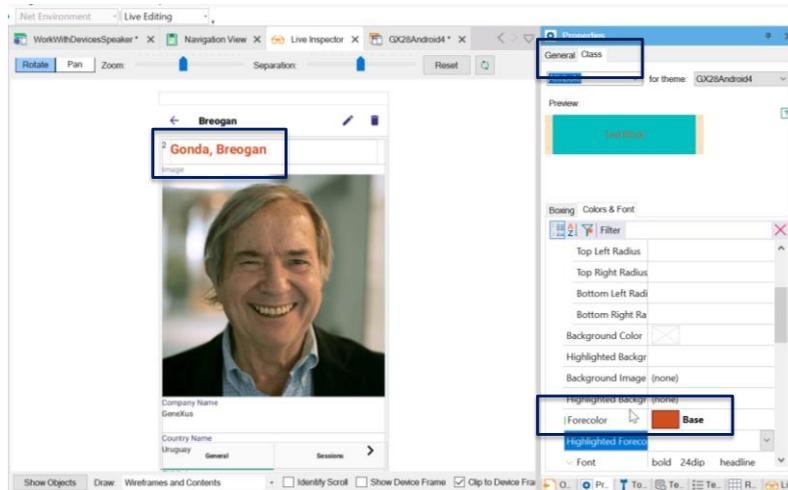


Acá podemos ver el List de Oradores, y vemos que todo lo que vemos en el emulador o en el dispositivo se va a ver reflejado en el IDE dentro de la pantalla Live Inspector. Vamos a acceder a un Orador, si seleccionamos un control en Live Inspector podemos ver sus propiedades, por ejemplo vamos a SpeakerFullName y cambiemos ahora el valor de Label Position a Left. y ahí vemos ya como queda, no fue necesario grabar ni compilar, ni ejecutar nuevamente la aplicación. si ahora ponemos el valor Bottom nuevamente lo vemos en forma inmediata. Vamos a dejar el valor None.

Seleccionemos ahora el atributo SpeakerId, le ocultaremos su etiqueta también, en realidad vamos a dejarlo oculto al atributo directamente. ponemos la propiedad visible en False y vamos a poner la propiedad Invisible Mode en Collapse Space para que cuando este oculto el control no ocupe lugar en la pantalla. Ahora vamos a cambiar la clase al atributo SpeakerFullName, vamos a usar la clase AttributeSessionTitle, veamos que ya tenemos el cambio en el IDE.

Editing Object & Theme with Live Inspector

Live Editing



Además de poder cambiar las propiedades de los atributos, en la solapa Class podemos directamente editar el Tema, vamos a Colors & Fonts, y usemos un color mas llamativo, elijamos dentro de la paleta el color Base, listo, ya podemos ver el cambio.

Vamos a deshacer este cambio ya que fue solo para mostrarles a ustedes como usarlo.

Vamos a asignar ahora la clase AttributeSpeakerNameT que es la clase que nos indico el diseñador.

Vamos a ir ahora al objeto directamente, aquí también podemos hacer cambios y verlos reflejados instantáneamente, de hecho vean que ya tiene los valores que pusimos recién en Live Inspector.

Otra cosa que podemos hacer es por ejemplo mover un control, llevemos el atributo SpeakerImage arriba de SpeakerFullName y lo dejamos sin Etiqueta.

Vamos a quitarle la etiqueta a CompanyName y le asignaremos la clase AttributeSpeakerBioT.

Ahora a CountryName, hacemos lo mismo, quitamos la etiqueta y asignamos la misma clase.

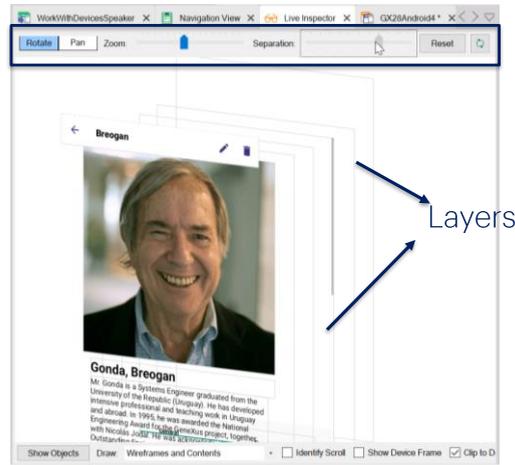
Llevemos el atributo SpeakerCVMini arriba, debajo de SpeakerFullName. también quitamos la etiqueta y asignamos la misma clase.

Marquemos ahora los atributos SpeakerAddress, SpeakerEmail y SpeakerPhone,

manteniendo presionado Control podemos seleccionar varios controles.
Le asignamos Label Position TOP, veamos como se ve en el emulador.
OK nos falto asignar la clase, asignamos la clase, la misma clase que usamos antes.
Vamos a ver de nuevo como quedo, OK. ahora si.
Vamos a agregar un titulo, vamos a ir a Tool Box y agregamos un control de tipo TextBlock
arriba de SpeakerAddress, vamos a poner de Caption Contact y la clase
TextBlockTitleWidgetM, revisemos, esta perfecto.
Ahora si recién vamos a grabar los cambios porque quedamos satisfechos con el diseño.

Live Inspector controls

Live Editing



Controls

Layers

Ahora vamos a ver algunas otras particularidades de la pantalla Live Inspector. por ejemplo si hacemos click y movemos el mouse podemos rotar la imagen en cualquier dirección, y fíjense que se ven unas capas cuando hacemos esto. Estas capas nos permiten seleccionarlas de una manera mucho mas sencilla, ya que en general están superpuestas, cuando son secciones o tablas. con este control podemos aumentar o disminuir el espacio entre las capas, con Reset volvemos a la visualización por defecto, con este control podemos aumentar o disminuir el Zoom sobre lo que estamos viendo y si seleccionamos Pan, en lugar de Rotate, podemos mover el contenido con el mouse, sin Rotarlo. Bien, esto fue todo, Vamos a volver ahora a la presentación así revisamos otras cuestiones relacionadas a Live Editing.

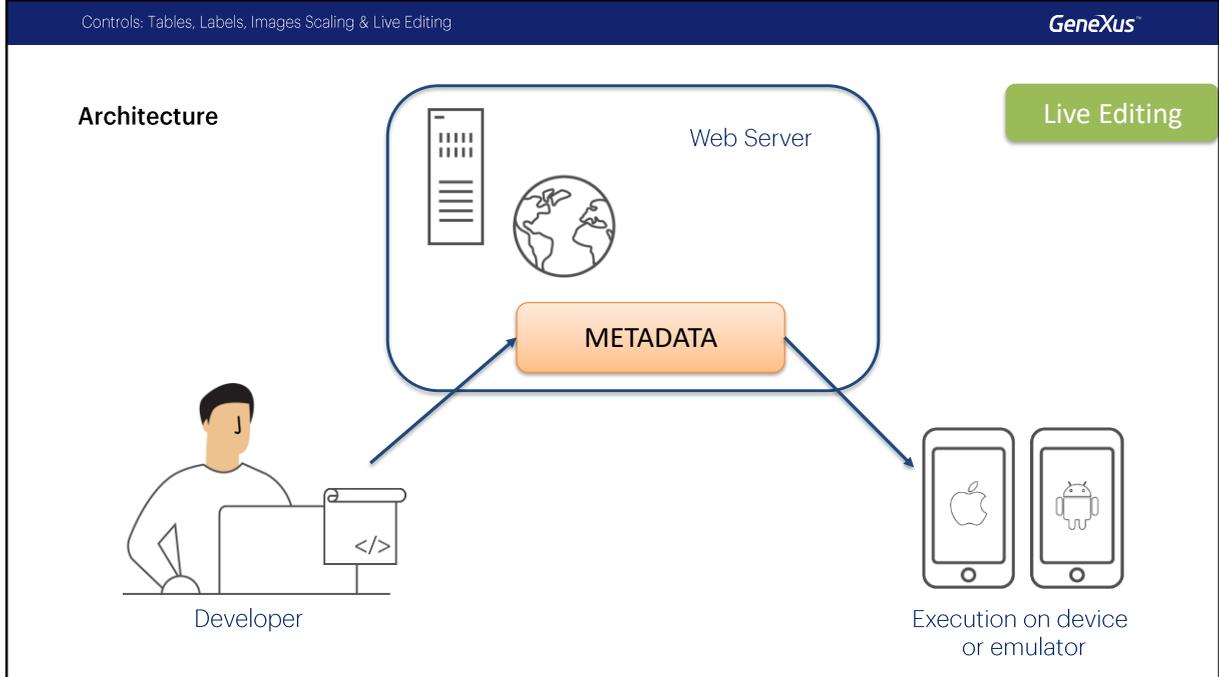
Live Editing Restrictions

Live Editing

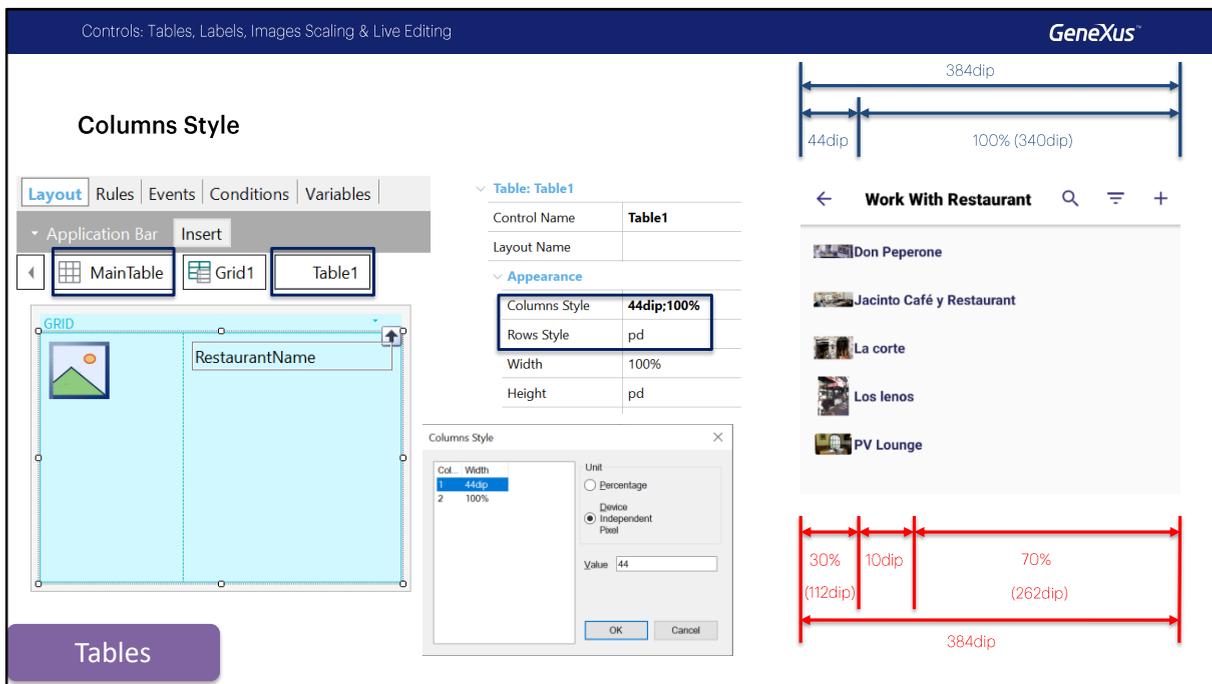
- Object Menu for Smart Devices is not supported.
- Server-side events (such as Start, Refresh and Load) cannot be tested with Live Editing because they are executed in the server-side (not in the device).
- Live Editing only works with the compiled application. Knowledge Base Navigator (KBN) is not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

Las restricciones para el uso de Live Editing son:

- No funciona en objetos de tipo Menu for Smart Devices.
- Las modificaciones sobre eventos Server Side (del lado del servidor) como Start, Refresh y Load no surgen efecto, por lo que es necesario re compilar la aplicación.
- Solo funciona cuando tenemos una aplicación compilada, no podemos usarlo cuando prototipamos con KBN.
- No funciona en Windows 7.



Como funciona esta característica? cuando usamos Live Editing el Servidor quedará “escuchando” los cambios que hagamos y replicará estos cambios sobre la Metadata a la que accede el dispositivo móvil (en nuestro caso el emulador) y que se utiliza para dibujar la pantalla en el dispositivo y definir su comportamiento.



Vamos a continuar con el diseño, ahora centrando nuestra atención en las Tablas. En esta sección vamos a utilizar el WorkWithDevicesRestaurant, si aun no han aplicado el patrón a esta transacción pueden hacerlo, también hemos agregado una opción al menú para acceder al mismo, como hicimos con Oradores y Países.

El trabajo con las tablas será central en las aplicaciones para Smart Devices. Todo layout, así esté vacío, tiene una tabla raíz, llamada MainTable. A su vez, todo grid va a definir una tabla para contener los controles de cada línea a ser cargada. En este caso, la tabla tiene dos columnas y una fila. Observemos las dos propiedades: **Columns Style** y **Rows Style**, que permiten definir el tamaño que ocupará cada columna y cada fila de la tabla.

Observemos las columnas. Sus tamaños se pueden especificar en dos unidades: porcentaje o dips (**D**evice **I**ndependent **P**ixel).

El Device Independent Pixel corresponde a una abstracción de un pixel que luego una aplicación convierte a pixeles físicos, lo que permite escalar a diferentes tamaños de pantalla. El dip para cada plataforma tiene diferente número de píxeles.

Los porcentajes son relativos al valor que resulta de restar del ancho total, los valores fijos (en dips). Así, si el ancho total es de 384 dips, y hay una columna de 44 dips, el valor de

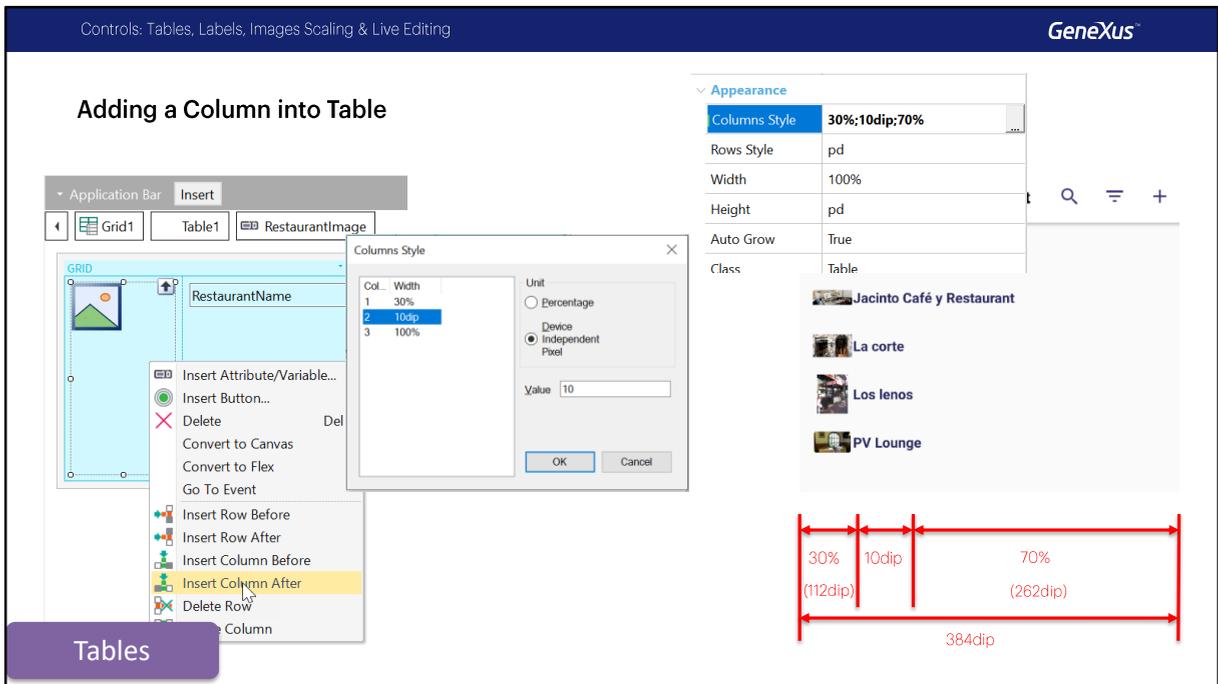
100% para la siguiente significa que esa segunda columna ocupará el 100% restante, es decir, $384 - 44 = 340$ dips.

Si tuviéramos tres columnas, la primera con 30%, la segunda con valor fijo de 10 dips y la tercera con 70%, los valores que asumirán la primera y la tercera se obtienen de aplicar esos porcentajes al valor resultante de sustraer la suma de los valores fijos (aquí sólo uno, 10 dips) del ancho de la tabla. La segunda columna con 10 dips puede usarse como espacio en blanco entre la primera y la tercera columna.

Debido a esto, la cantidad de espacio disponible para la primera y tercera columnas es de: $384 - 10 = 374$ dips, por lo tanto la primera columna tendrá el 30% de 374 que equivale a 112 dips y la tercera el 70% restante, que son 262 dips.

Demo: Working with Tables

Vamos a ver todo esto en GeneXus.



Vamos a acceder al objeto WorkWithdevicesRestaurant y vamos a ir al List.

Acá podemos ver el Main Table del cual hablamos recién y dentro de la grilla podemos ver la tabla con el nombre Table1 que posee actualmente 2 columnas, una de 44 dips y otra de 100%, tal como vimos en la presentación.

Entonces, vamos a insertar una columna adicional usando el botón derecho sobre la tabla del grid y la agregamos después de donde estamos parados, RestaurantImage. Vamos a especificar el valor de Columns Style como habíamos visto, vamos a poner 30% para la primer columna, 10 dip para la segunda y 70% para la tercera, vamos a ver la aplicación en ejecución así vemos el resultado, estamos usando Live Editing así que directamente podemos verlo, y acá vemos que cambió, ahora tenemos un espacio entre el nombre y la imagen.

Adding an Attribute into Layout

The screenshot illustrates the GeneXus IDE interface during the process of adding an attribute to a layout. The main workspace shows a 'GRID' control with a 'RestaurantName' label. The 'Insert' application bar is active, and the 'Attribute/Variable' control is being added to the grid. The 'Live Inspector' panel on the right shows the properties for the 'RestaurantAddress' attribute, including its class, visibility, and appearance settings.

Attribute/Variable: RestaurantAddress

Attribute	RestaurantAddress
Readonly	True
Appearance	
Label Position	None
Class	Attribute
Visible	True
Invisible Mode	Keep Space
Enabled	True
Format	Text
Invite Message	Restaurant Address

Tables

Vamos a agregar el atributo RestaurantAddress a la grilla, vamos a quitarle su etiqueta (colocamos Label Position en None), si vemos ahora en Live Inspector no vemos el cambio, esto es porque estamos haciendo un cambio, una modificación, sobre un evento Server Side (el load) entonces es necesario volver a compilar la aplicación, por lo que presionamos Run nuevamente.

Rows Style

Application Bar Insert

Grid1 Table1 RestaurantAddress

GRID

RestaurantName

RestaurantAddress

Table: Table1

Control Name	Table1
Layout Name	

Appearance

Columns Style	30%;10dip;70%
Rows Style	pd;pd
Width	100%

Rows Style

Row	Height
1	pd
2	pd

Unit

- Percentage
- Device Independent Pixel
- Platform Default

Value: 0

OK Cancel

Tables

“Using the best value depending on the platform and context”

Mientras se genera y compila la aplicación vamos a revisar las filas de la tabla, vemos que se agrega la unidad pd, Platform Default. Este valor difiere de plataforma en plataforma, y para una misma plataforma, también depende del contenido de la celda, por ejemplo, de si el campo tiene label o no, y si tiene, si la etiqueta será desplegada arriba o a la izquierda, o si estamos en un layout en modo Edit o en modo View.

El valor pd Corresponde a: “Usar el Mejor Valor Dependiendo de la Plataforma y el Contexto”.

Por ejemplo, para Android, con Label Position = Top, corresponde a 64 dips, mientras que en iOS a 53 dips.

Vamos a ver como quedo, bien, ya vemos el Address, vamos a cambiarle la clase, vamos a crear una clase AttributeRestaurantAddress y vamos a configurar la clase, sacamos Autolink (False), no queremos que tenga el icono para ir a la dirección, y vamos a poner en Font Weight Bold, listo.

Ya tenemos el diseño, grabamos y seguimos con la presentación.

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.

Vamos a ver algunas propiedades de las imágenes, específicamente los distintos valores que disponemos para el escalado de las imágenes, estos son:

No Scale: se va a respetar el tamaño original de la imagen independientemente del tamaño del área donde esta el control.

Tile: en este caso las imágenes no se escalan y se repetirá la imagen hasta llenar el tamaño del control.

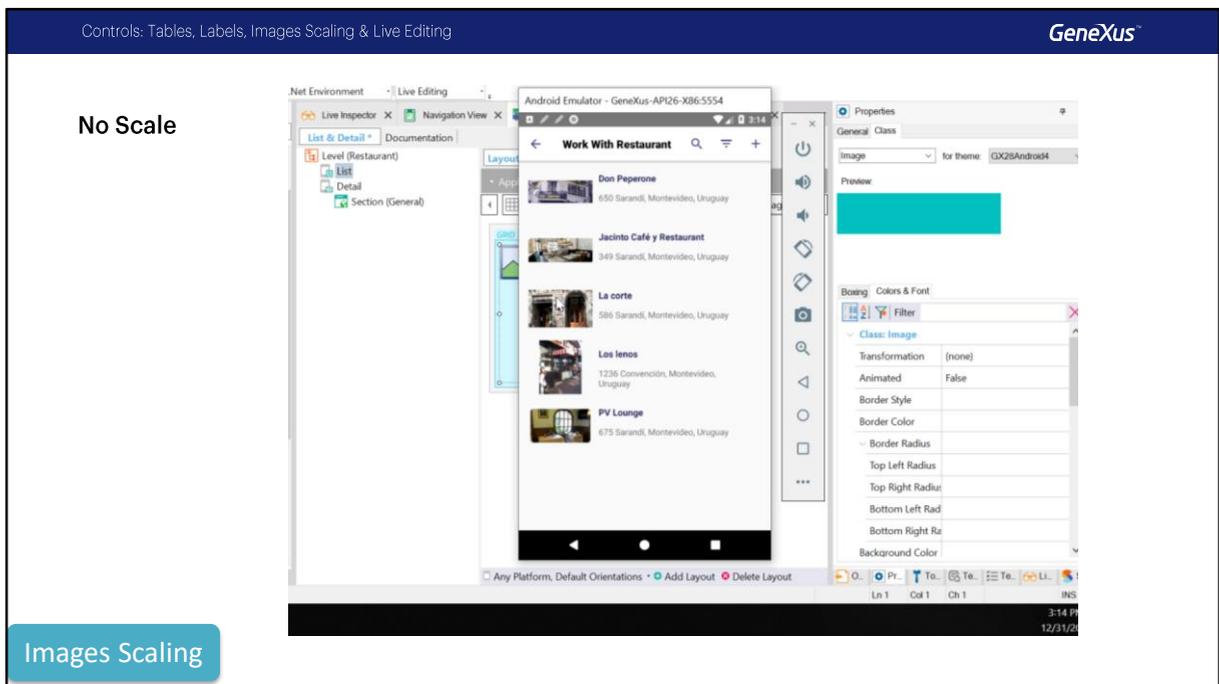
Fill: la imagen escalara en alto y largo hasta ocupar el tamaño del área donde esta el control.

Fill Keeping Aspect Ratio: es similar a Fill pero la imagen siempre va a mantener las mismas proporciones de ancho y alto, de esta forma la imagen no se deforma. Por ejemplo si el tamaño de la imagen es de 100x200 px y el tamaño del área del control es de 50x50, la imagen se convertirá a 50x100 px.

Fit: la imagen escalara en alto y ancho de modo de que se pueda ver en forma completa y manteniendo las proporciones y aspecto. Por ejemplo si la imagen es de 100x200 px y el tamaño del área del control es de 50x50, la imagen se convertirá a 25x50. Este es el valor default que se asume.

Demo: Images Scaling

Vamos nuevamente a GeneXus y veremos esto aprovechando Live Editing.



Así es como se está viendo actualmente el List de Restaurants, fíjense que las imágenes no están quedando muy bonitas y esto es debido al escalado de las imágenes, vamos a tratar de arreglar este panel.

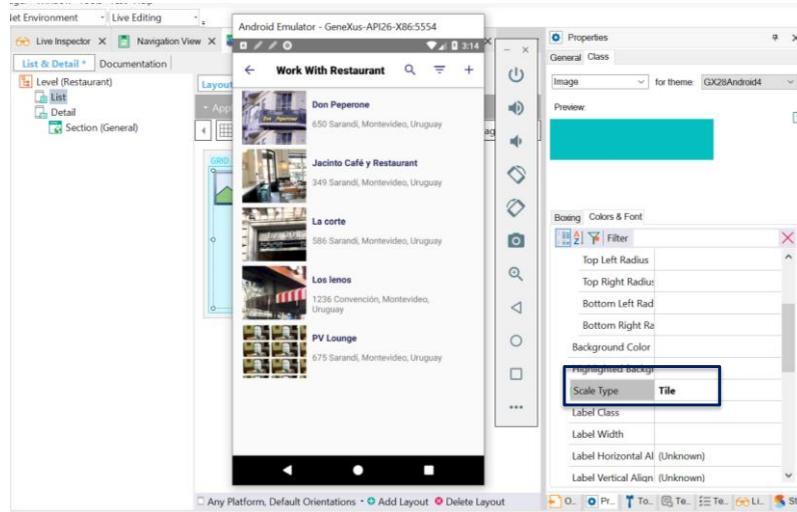
Lo primero que vamos a hacer es hacer que esta imagen ocupe 2 filas, esto es una propiedad Row Span del control, y vamos a indicar que ocupe dos filas, vamos a borrar la columna que se agregó automáticamente.

Ahora fíjense que nosotros teníamos una columna de 30%, otra de 10 dips y 70% la restante.

Vamos a ver como va quedando, fíjense que igual todavía las imágenes se deforman un poco, esto es justamente por la propiedad Scale Type de las imágenes, la clase de RestaurantImage es la clase Image, y vamos a ver que en Scale Type no tenemos ningún Valor.

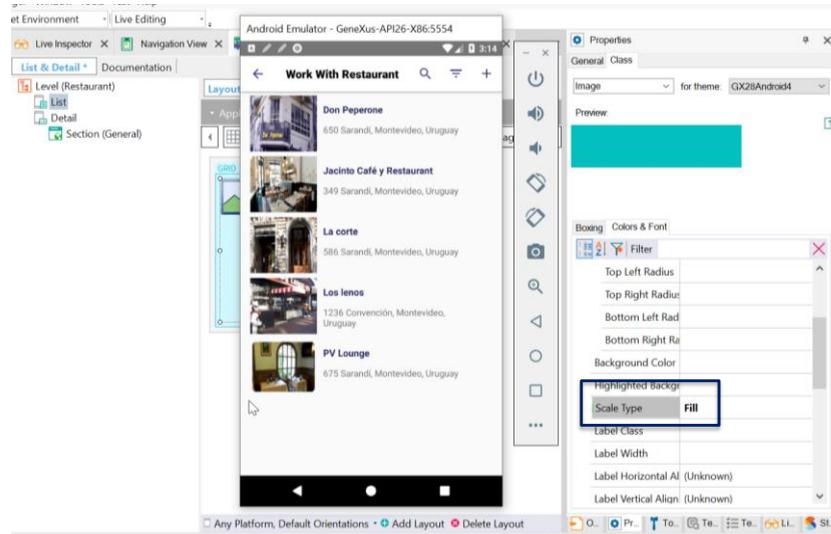
Vamos a ir cambiando estos valores para ver como se van modificando en ejecución. NO Scale, vemos que sigue igual.

Tile



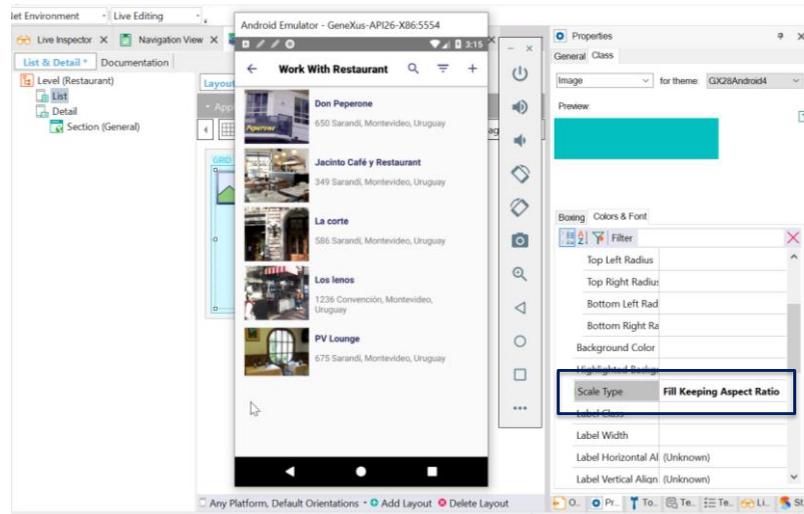
Images Scaling

En Tile, no vamos a apreciar este efecto. Bueno, en este caso si (la ultima imagen) esta imagen es mas pequeña que el tamaño (del área del control) por lo tanto la imagen se repite para rellenar el espacio disponible.

Fill**Images Scaling**

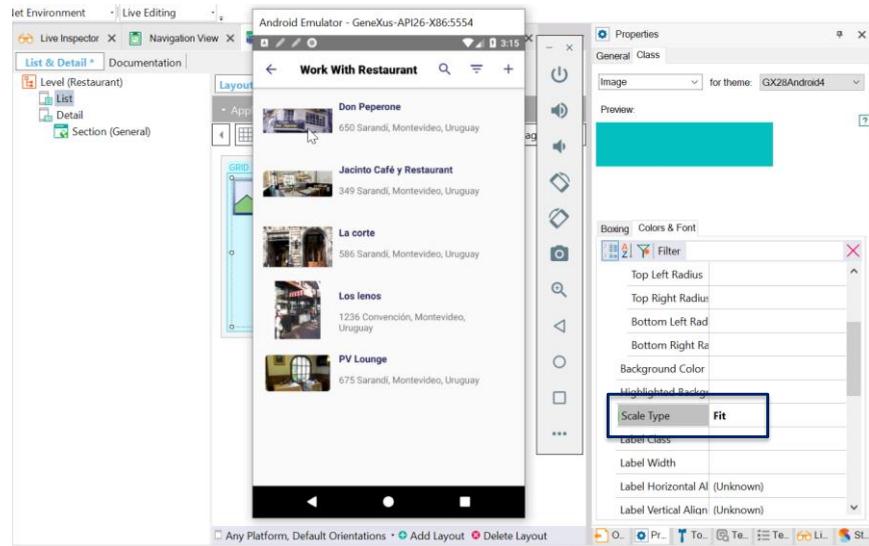
Vamos a ver con Fill, fíjense que en este caso la imagen rellena todo el espacio disponible pero se deforma.

Fill Keeping Aspect Ratio



Images Scaling

Vamos a ver Fill Keeping Aspect Ratio, fíjense que ahora si, la imagen rellena todo el espacio disponible pero se mantiene el aspecto de la imagen.

Fit

Images Scaling

Y Fit, que es el ultimo valor que nos quedaba, vean que rellena pero no va a ocupar todo el espacio disponible porque mantiene la proporción cuando rellena pero no lo hace completando todo el espacio disponible.

Vamos a dejar entonces el mejor que se veía que era Fill Keeping Aspect Ratio. Perfecto.

Label Position	Live Editing	Tables	Images Scaling
<ul style="list-style-type: none">• Platform Default• None• Right• Left• Top• Bottom• Float	<ul style="list-style-type: none">• Live Inspector• Using Restrictions• Architecture	<ul style="list-style-type: none">• Columns Styles• Rows Styles• Measure Units	<ul style="list-style-type: none">• No Scale• Tile• Fill• Fill Keeping Aspect Ratio• Fit

Vamos a repasar todo lo que vimos en este video.

- Vimos las distintas opciones para Label Position: Platform Default, None, Right, Left, Top, Bottom y Float. Este ultimo solo cuando es en modo edición.
- Aprendimos a utilizar Live Editing, usamos la herramienta Live Inspector y hablamos de las restricciones para su uso y la arquitectura de esta herramienta.
- Sobre Tablas, vimos como especificar los estilos de las columnas y de las filas. Vimos las unidades de medida DIP, Porcentajes y el valor pd, Platform Default, para Filas.
- Por ultimo, hablamos sobre el escalado de imágenes y vimos los distintos valores: No Scale, Tile, Fill, Fill Keeping Aspect Ratio y Fit.

GeneXus™

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