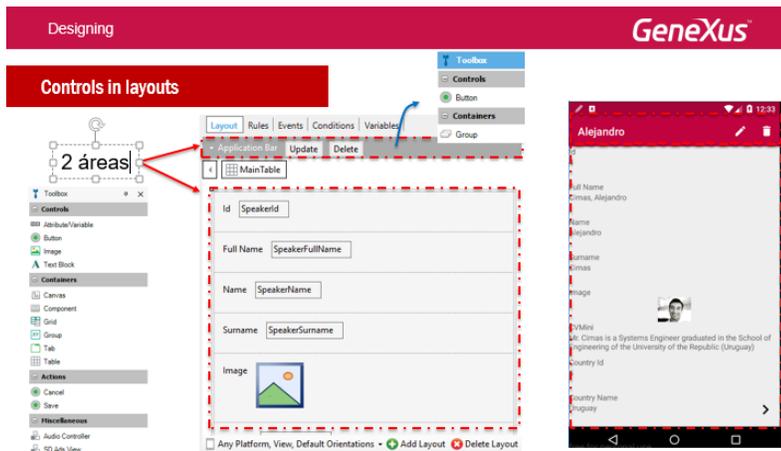


Ahora nos vamos a dedicar a las particularidades que asumen los controles en los layouts, todo esto respecto al uso que ya conocíamos para web panels; fundamentalmente los aspectos que se indican acá, empezando por el primero.



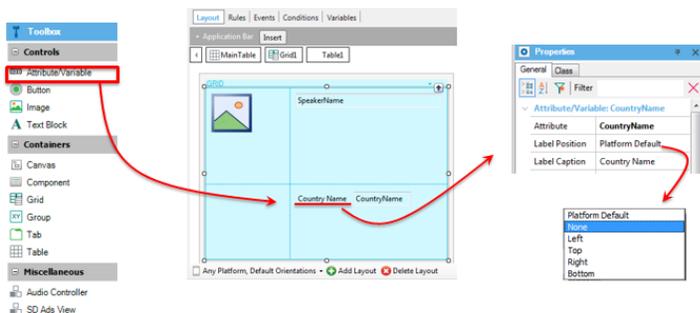
1 Dos áreas componen un layout:

El área conocida como **Application bar**, donde se pueden colocar botones (que puedan tener asociadas imágenes o no tenerlas), y que van a corresponder a acciones que se quieran realizar sobre los datos o llamar a pantalla o lo que sea o contenedores de grupos de acciones (Group control), para agrupar varias acciones y ofrecerlas, por ejemplo, como menús desplegables.

Y por otro lado vamos a tener el área del layout propiamente dicha, que nos ofrece la toolbox para poder insertar los controles (si se trata del nodo Detail vamos a ver después qué van a aparecer además de estos que estamos viendo acá el placeholders para las distintas secciones).

Bueno, nos vamos a concentrar en esta segunda área ahora y dejaremos la primera para más adelante.

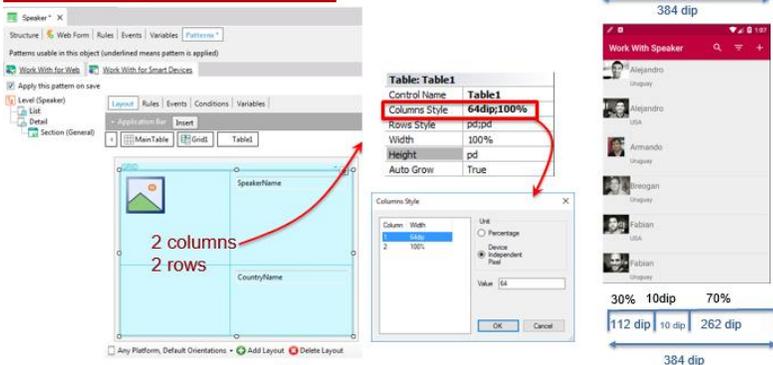
Controls in layouts: atts/vars



Acá estamos viendo el List del Work with de Speakers. vemos que por default se mostraba SpeakerImage y SpeakerName. Y vamos agregar un atributo, el country el país del orador.

Para los controles de atributos o variables, tenemos la propiedad **Label Position** que va a permitir especificar la posición que ocupará la etiqueta del atributo la lebel o variable que es un atributo la posición que va a ocupar o etiqueta. Cada plataforma tiene su default (en Android, por ejemplo, es Top, lo que significa que la etiqueta va a aparecer arriba del control atributo y no como lo estamos viendo aquí, que es a la izquierda, porque de algún modo hay que mostrarlo y esta pantalla es genérica, entonces no es una pantalla para Android especialmente. Si seleccionamos el valor "None" entonces la etiqueta no se mostrará., vamos a verlo en GeneXus.

Controls in layouts: table



Vamos a el work with speaker, vamos al list y bueno acá vemos entonces los dos controles que tenemos por ahora, en el grid del layout, vamos a la tull box y vamos a insertar un nuevo atributo arrastrando, lo elegimos es country name y vemos que primero que nada agregó una nueva fila y columna a la tabla del grid y por otro lado si vamos a las propiedades vemos que tenemos la propiedad level position con el valor Platform default que si ahora ejecutamos vamos a ver que para android es arriba.

Vamos a cambiarlo a none para que no aparezca la etiqueta, guardamos, bueno y ejecutemos para ver el cambio.

Hacemos ram sobre el objeto dashboard, vamos a abrir el emulador que quedo abierto de la ejecución anterior, de hecho si nos apuramos antes de que cargue el nuevo pk vemos que solamente estaban saliendo foto y nombre y ahora en el nuevo va a tener que aparecer el país, bueno apreciamos todo lo que está demorando hacer la compilación que en este caso en realidad ahora lo dije así pero en verdad no vamos a tener más remedio porque estamos agregando un nuevo control un nuevo atributo y eso va a tener que ir a la base de datos a buscar la información, así que en este caso concreto no podemos evitar todos estos pasos de compilación y subida a la nube, pero vamos a ver en qué caso si podemos

evitarlo, en ratito nada más, entonces vemos que acá está cargando y si vamos a speakers ahora está apareciendo el país, volvemos a las ppt, bien..

El trabajo con las tablas va a ser central en las aplicaciones para Smart Devices. Todo layout, así esté vacío, tiene una tabla raíz, llamada MainTable. Que podemos verla acá estamos posicionados sobre el grid, sobre la tabla del grid y nos va a mostrar acá el control sobre los que estamos posicionados y todas las jerarquías, el padre y donde está inserto ese padre todo control está inserto en la Main Table.

A su vez, todo grid va a definir una tabla, decíamos, esta tabla de acá para contener los controles de cada línea a ser cargada, de cada elemento de ese grid que va a ser cargada de la base de datos. En este caso, la tabla tiene dos columnas y dos filas. Observemos las dos propiedades: **Columns Style** y **Rows Style**, que van a permitir definir el tamaño que ocupará cada columna y cada fila de la tabla.

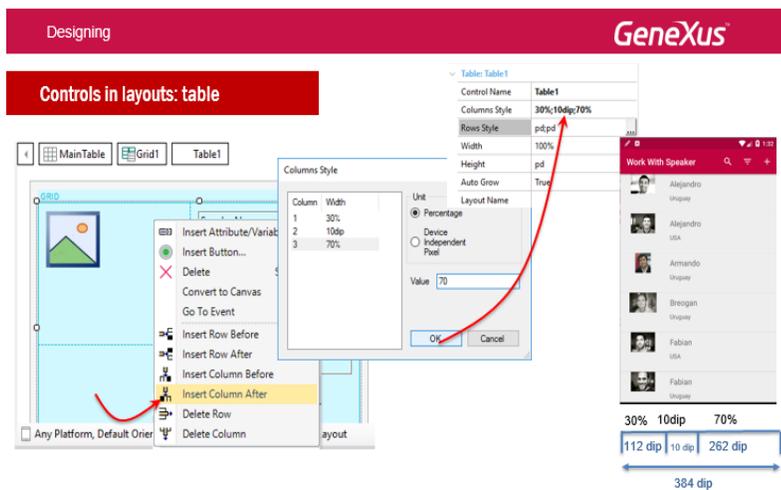
Si observamos las columnas vemos que sus tamaños se pueden definir o especificar en dos unidades: porcentaje o dips (**device independent pixel**).

Esta unidad corresponde a una abstracción de un pixel que luego una aplicación convierte a píxeles físicos, lo que va a permitir escalar a diferentes tamaños de pantalla usando tamaños uniformes, el dip.

El dips 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, el ancho de la pantalla y hay una columna de 64 dips, como vemos acá el 100% que va a ocupar la columna restante va a ser el que resulte de restar este ancho total esta cantidad de dips y ese va a ser el tamaño entonces de esa columna, en este caso 320 dips del ancho.

Si tuviéramos tres columnas, lo mostramos acá, la primera con 30%, la segunda con valor fijo 10 dips y la tercera con 70%, los valores que asumirán la primera y la tercera que están en porcentaje se van a obtener de aplicar esos porcentajes al valor resultante de sustraer de restar la suma de los valores fijos (aquí sólo uno, 10 dips) al total que en este caso es 384, o sea $384 - 10 = 374$ entonces la cantidad de dips que va a ocupar la primera columna va a ser el 30% y la última va a ser el 70% de ese valor.

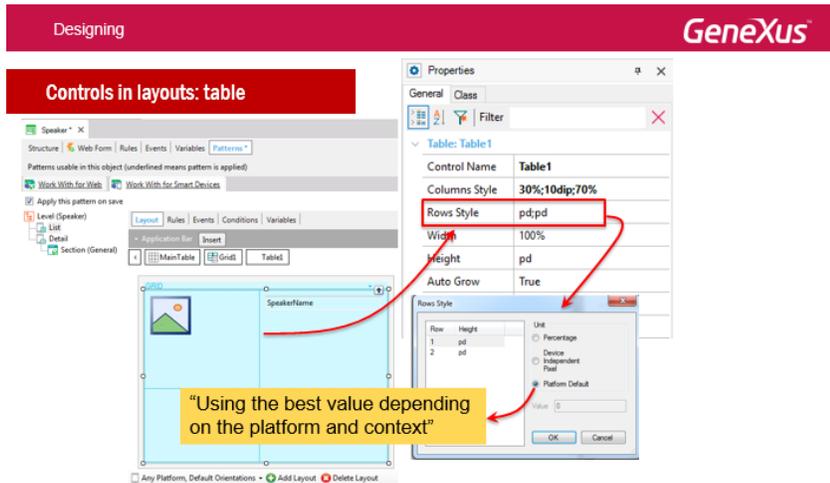


Entonces vamos a hacer esto, vamos a agregar una columna nuestra tabla tenía dos columnas vamos a agregar una tercera de 10disp para dar un espacio, un aire, como podemos ver acá, un aire entre la primera columna y la tercera para que no queden los datos pegados, como quedaban en este caso de la aplicación, vamos de vuelta a GeneXus, vamos al world with del speakers y acá nos posicionamos en cualquier lado y le damos insert column after y ahora vemos que tenemos 3 columnas, nos posicionamos en la tabla y modificando sus propiedades vemos que los tamaños de la columna asumen la primera

64dips, la segunda 50% restante y la tercera el 50% restante, entonces vamos a modificar esos valores para que la primera asuma el 30% la segunda un tamaño fijo de 10dips y la tercera el 70% restante.

Bien perfecto entonces queremos probar, cómo hacemos? por ahora si seguimos a la la vieja usanza, haríamos botón derecho otra vez sobre el dashram, run, a menos que tengamos como start up option ahí podemos dar F5 directamente. Otra vez vamos a poner en edición mostrar y este si va a ser un cambio para el que no necesitamos compilar la aplicación y podemos verlos instantáneamente y no esperar todo este tiempo.

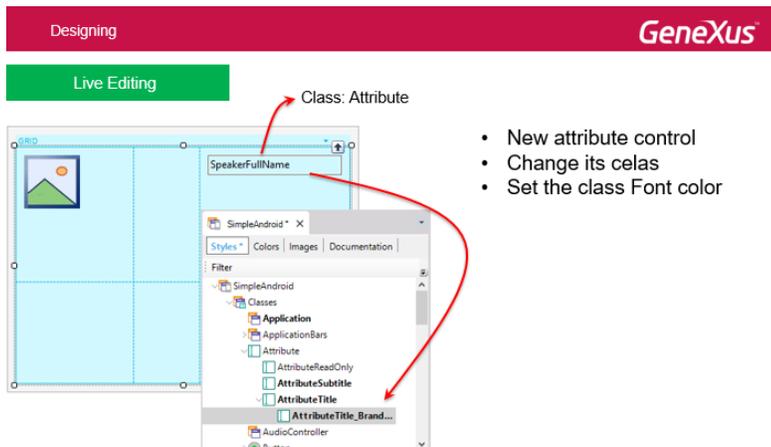
Bueno ahí está apareciendo la aplicación así que vamos a speaker,s ahora si vemos este aire entre un dato y el otro, esto es lo que acabamos de ver perfecto.



Por otro lado también si ahora observamos las filas, vemos que se agrega la unidad pd, Platform Default, que no estaba en el caso anterior.

Este valor platform default difiere de plataforma en plataforma justamente por eso es un default por plataforma, y para una misma plataforma, también depende del contenido de la celda, por ejemplo, de si el campo tiene etiqueta label o no, y si tiene, si será desplegada arriba o a la izquierda, o si es en un layout en modo Edit o View todo eso va a ser variable.

El valor platform default Corresponde a: "Using the best value depending on the platform and the context". Por ejemplo, para Android, con Label Postion = Top, corresponde a 64dips, mientras que en iOS a 53.



Ahora sí llegamos al momento de simplificarse en los tiempo de prueba de cambios de diseño y de prueba de los cambios este y entonces vamos a ver la herramienta live editing.

Entonces hasta ahora cada vez que queríamos probar los cambios de los tamaños de las filas o las columnas de las tablas, como vimos debíamos volver a especificar y generar la aplicación y compilar y todo eso demoraba su tiempo.

Acá vamos a introducir esta herramienta que va a permitir realizar cambios de diseño y algunos cambios de comportamiento y ver instantáneamente esos cambios sin siquiera grabar el cambio de manera tal que podemos deshacer inmediatamente sin espera, sin ningún tiempo.

Vamos a verlo con un ejemplo. Entonces qué vamos a hacer? vamos a observar primero que nada el atributo que teníamos en el grid de trabajar con speakers que era SpeakerName, vamos a ver que ese speakers name tenía asociada la clase AttributeTitle por eso salía más grande ese valor del atributo el nombre del speakers salia más grande que el resto, si lo vemos en en el emulador, podemos apreciar que el justamente el nombre está saliendo mucho más grande, entonces si vamos a observar en speaker vemos la propiedad asociada a la clase AttributeTitle podemos ver que en esa clase entonces tenemos este tamaño y eso es lo que lo está haciendo más grande.

Bueno vamos a cambiar este atributo, vamos a borrarlo, este que tenía la clase AttributeTitle y vamos a querer poner el atributo en su lugar, insertar el atributo speaker full name, vamos a sacarle el level ponerlo en none, perfecto.

Si ahora observamos cómo estamos insertando un atributo nuevo, el patters es el que había colocado la clase AttributeTitle al attribute anterior, ahora tiene la clase Attribute que tiene el valor estándar que se va a ver igual que el atributo country name.

Entonces lo primero que vamos a querer hacer es cambiarle su clase, por ejemplo pensemos que en vez de ponerle como título podríamos querer cambiarle para que tenga también la forma que tenía antes pero lo que vamos a querer hacer no es especialmente o no solo eso, sino que además que aparezca con el color de la marca, que aparezca entonces con ese color rojizo que tenía la marca.

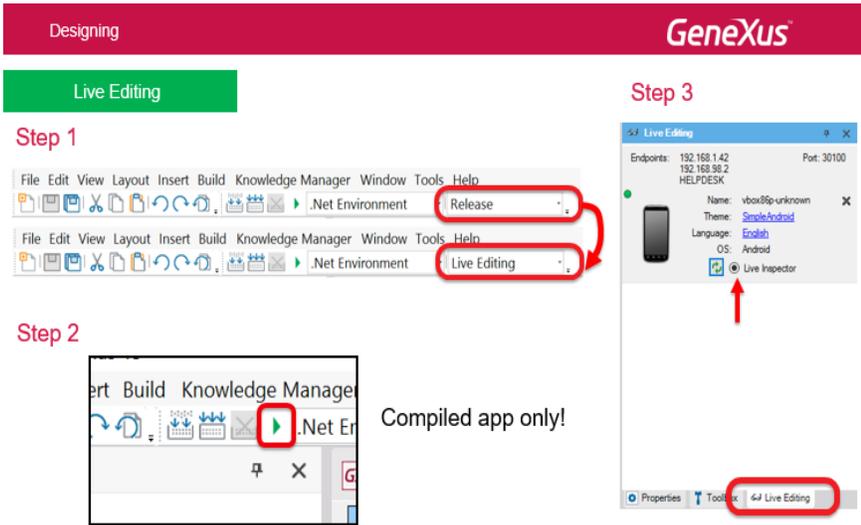
Entonces para eso vamos a grabar el atributo este, si lo mostramos vamos a dejar especificando y generando porque cambie un atributo y en ese caso no tengo más remedio que volver a compilar porque va a tener que ir hasta la base de datos a buscar esa información nueva en el ran y mientras voy haciendo el ran voy a mirar en el tema circul Android voy a agregar bajo la clase Attribute AttributeTitle que vemos, bueno ya estaba agregada , ya agregue, en realidad para evitar demorar, vean que agregue la clase ésta AttributeTitlebrand que va a tener como four color el color del brand, y además vean que también acá agregue la clase attribute traks y ya le fui dando colores a las distintas subclases, lo mismo hice a nivel de las tablas, vean que agregue esta clase table track que tiene, si abrimos y expandimos tiene estas subclases, por ejemplo blue para que aparezca azul el color de fondo cuando está seleccionada o que aparezca blanco y azul , etc.

Esto lo vamos a ver más adelante, que va a ser lo que le va a dar ese aspecto de diseño a la aplicación Android tan lindo que va a ir cambiando el color de acuerdo a la conferencia de la que se trate, los traks.

Bueno, entonces acá vemos lo que decíamos en el brand color, que no sé por qué no lo está mostrando, debería estar mostrando el color, en ese caso vamos ahora si a speakers ya vamos a ver que levantó en el emulador, vemos que entonces ahora está el fullname que está del mismo tamaño que el country name, perfecto.

Entonces qué queremos hacer decíamos? cambiarle la clase a este atributo para que ahora suma la clase nueva que ingrese pero para hacer esto que haríamos cambiaríamos la clase haríamos otra vez un ran sobre el dashboard, pero no quiero eso, quiero que automáticamente hacer el cambio y eso se vea reflejado en el emulador, que ya pueda ver el cambio sin hacer nada más sin ni siquiera grabar, para eso decíamos contamos con la herramienta live editing.

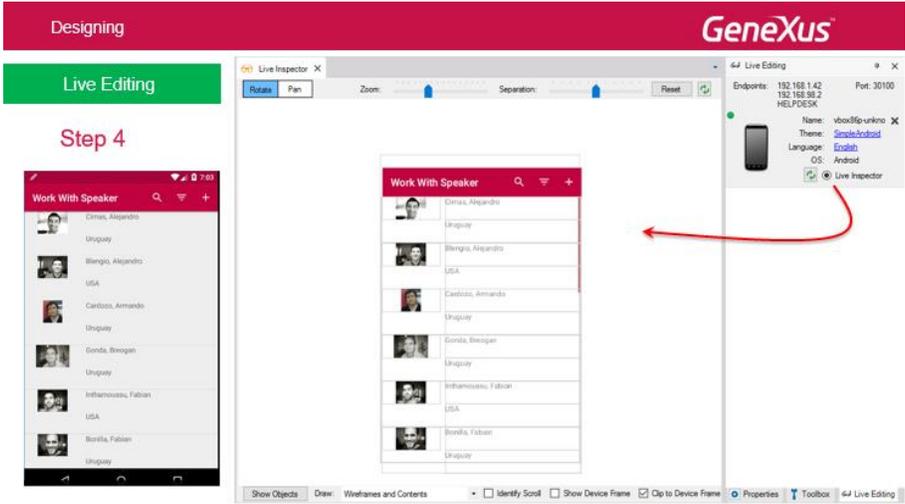
Cómo accedemos a esa herramienta? cambiándonos acá de la pestaña release, del valor release de este combo a Live Editing. Vemos que al hacerlo aparece esta pestaña de acá abajo con esto que en realidad por ahora no tiene nada, ahora sí, yo tengo que otra vez una vez que me cambie tengo que compilar la aplicación para que live editing empiece a escuchar a esa aplicación compilada en el dispositivo y se conecte, pueda conectarse y escuchar lo que aparece en el dispositivo, live editing mientras esperamos que termine la compilación.



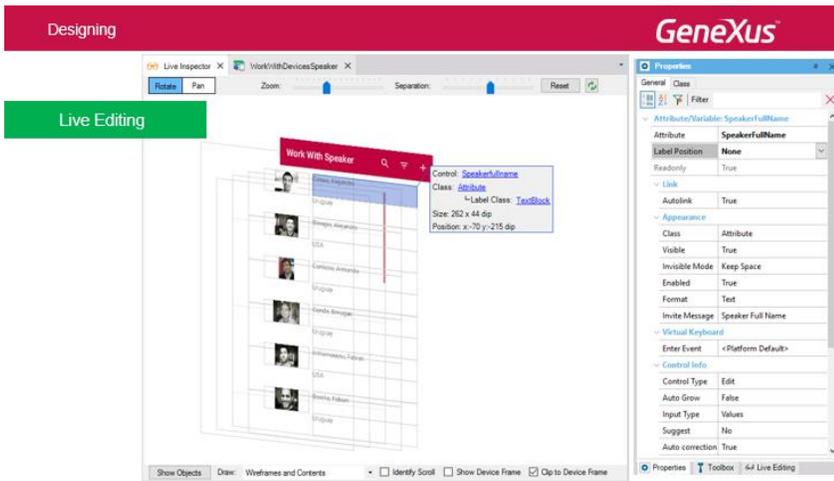
Live Editing es una herramienta de GeneXus que permite realizar modificaciones en themes, languages, layouts y eventos de usuario de nuestros paneles y ver los cambios instantáneamente en la aplicación en ejecución, sin tener que especificar, generar y compilar. Y sin siquiera tener que guardar los cambios. Ya lo repetí varias veces pero es muy importante eso.

De esta manera el servidor de Live Editing va a quedar “escuchando” los cambios que hagamos a la aplicación y va replicará estos cambios sobre la metadata a la que accede el dispositivo móvil (nuestro caso es el emulador pero podría ser un dispositivo real)

Esa metadata es la que se utiliza para dibujar la pantalla en el dispositivo y para definir su comportamiento entonces modificando la metadata va a poder hacer las modificación instantáneamente sin necesidad de compilar nada. Vamos a ver si ya levantó...

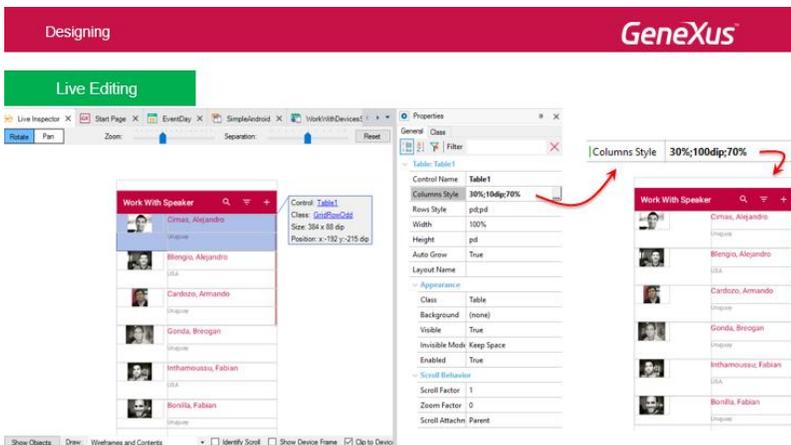


Vemos que ahora aparece en nuestro emulador la aplicación y vemos que está apareciendo esta entrada de acá que me está diciendo que está escuchando a este emulador, a la aplicación en este emulador, que tiene el tema simple Android que el lenguaje english y que el sistema operativo es Android.



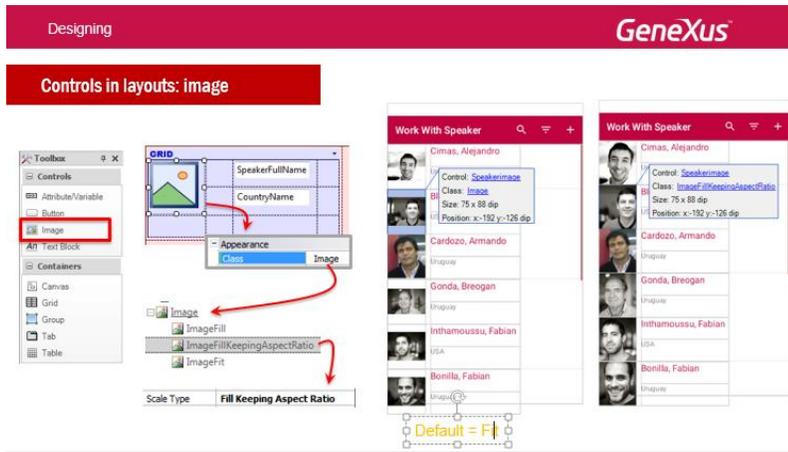
Qué es lo que quiero hacer? activar el live spectator, al hacer eso vemos que se nos abre otra pestaña de aquí que ahora parece no estar mostrando, nada porque el dashwor no es un elemento que sea inspeccionable modificable y sus cambios visibles en esta pantalla, pero si en el emulador elegimos speakers por ejemplo vemos que automáticamente se está reflejando en esta otra pantalla de acá, por ejemplo si hiciéramos tap sobre el orador Alejandro Cimas vemos como automáticamente se carga en la ventana de live inspector se carga la información que estamos viendo en tiempo real en el dispositivo, voy para atrás y vemos que otra vez aparece, como cosa interesante además nosotros podemos trabajar directamente sobre esta pantalla y cambiar aspectos de diseños automáticamente aquí y verlos automáticamente reflejados, esta pantalla nos permite hacer zoom sí agrandar o achicar y además con el mouse girándola podemos ver las distintas capas que componen el layout que estamos visualizando para poder trabajar mejor con él, por ejemplo separarlas más o no separarlas, tenerlas en su grado cero, si, entonces acá vemos por ejemplo el applicationbar haciendo clic vamos a las propiedades y vemos que se activan las propiedades del dash word acá tenemos la clase, el control application bar si por ejemplo me posiciono vamos a girarlo un poco o dejarlo incluso a plano, puedo resetear y incluso lo tenemos de forma plana, me posiciono sobre Cimas Alejandro, vemos que me aparece de qué control se trata, aparece activo acá en las propiedades del control, aparece la clase y la subclase, de manera tal que yo puedo modificar aspectos del control tanto a nivel de las propiedades y como de las clases y vamos a poder ver que automáticamente vemos estos cambios por ejemplo vamos a cambiarle la clase a ese control vamos a pasarla a esa que yo decía que había agregado que era atributetitle brand color, y automáticamente cuando salga del campo se va a ver el cambio.

Hice el cambio directamente en el screen y vean como se reflejó automáticamente en el Live Inspector que es lo mismo el Live Inspector está escuchando lo que sucede en el emulador entonces el cambio lo hace automáticamente vean todo el tiempo que ahorramos en este caso no?



Bueno y eso es lo que decíamos verdad? entonces podemos hacer estos cambios e incluso esto de cambiar el tamaño de las columnas y todo esto lo podríamos haber hecho automáticamente.

Por ejemplo, vamos a hacerlo ahora que no nos cuesta nada, posicionarnos acá en control en tabla 1 el que tiene las columnas y las filas y por ejemplo cambiar de 10 dips a 100 dips y cuando salimos vemos que automáticamente reflejo el cambio y yo no tengo porque grabar este cambio sino quiero y si no lo grabo entonces voy a volver al tamaño anterior.



Otra cuestión interesante tiene que ver con el uso de las imágenes, acá lo que vemos en estas imágenes de aquí es que hemos cambiado la propiedad Round span de la imagen para que ocupe dos filas en vez de una , si observamos la clase Image default no tiene configurado ningún valor para la propiedad scale type eso va a significar que asume el valor fit es decir que va a tratar de encajar en tamaño que tiene, si queremos cambiar ese comportamiento va a alcanzar con asociarle una clase que configure la propiedad scale type correspondiente, por ejemplo vemos acá esta es el default la propiedad fit y lo que vamos a querer es que la propiedad tenga el valor fil keepin aspect reader, de manera tal que la imagen se expanda pero que mantenga las proporciones, si, entonces se va a acortar de la manera que corresponde.

Vamos a tener que crear estas clases, que no existen, porque va a existir solo la clase Image, haciendo variar entonces el scale type.

Vamos a GeneXus, voy al tema, luego voy a la clase Image.

Bueno y otra vez, como yo ya importe el tema, y ya había hecho estos cambios, aparecen estas tres clases, pero en el tema default no van a aparecer, los tenemos que crear. Por ejemplo ésta: Image keeping Aspect Ratio que es una subclase de la clase Image y vean que lo único que configure fue la propiedad scael type.

Acá podemos ver, filtrar solamente por las propiedades que han sido modificadas de la clase y ahí la podemos ver simplemente, a la propiedad.

Bien perfecto, entonces lo que vamos a hacer es dos cosas, uno a la imagen decirle que se expanda para ocupar 2 filas y no solamente 1 y ahí vemos el resulta, bueno aca esta columna la vamos a tener que mover porque se nos corrió para allá la ponemos donde corresponda, eliminamos la última columna indeseada delit column, ahora vamos al live inspector, tenemos todo en orden, vamos a ponerlo todo para verlo mejor y lo que decíamos es que no está ocupando, la clase que tiene es la clase estandar Image, no está ocupando siendo rellena por el espacio, entonces vamos a modificar la clase a Image fill keeping Ratio y vemos cómo todo se acomodó de esa manera, otra vez si vemos en el emulador es lo mismo no, vemos como esta todo funcionando tal como deseábamos.

Y bueno los controles imágenes tiene esas posibilidades tipo de imagen que bueno dejamos para que ustedes vean pero en particulares tenemos was have fit para rellenar sin escalar proporcionalmente, para que rellene toda el área de pantalla, etc.

Controls in layouts: image

<u>No Scale</u>	Respects the original size of the image, independently of the control area size.
<u>Tile</u>	The image is not scaled. It is repeated horizontally and vertically to fill the control size.
<u>9 Patch</u>	The image must have the Scalable Image Property set to true. This images contain information about how they should be scaled.
<u>Fill</u>	The image is scaled in width and height in order to fill the whole size of the control area.
<u>Fill Keeping Aspect Ratio</u>	The image makes bigger or smaller in width and height in order to fill the whole 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.
<u>Fit</u>	The image scales in width and height in order 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.

Bien con esto terminamos entonces con el primero de los temas de controles de layout, vamos a pasar al siguiente, Multiples layouts per fit.